



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria
de Manresa



Treball Final de Grau

Control digital per a la levitació magnètica d'un eix, amb parametrització i anàlisi de dades mitjançant IoT.

Grau en Enginyeria Electrònica Industrial i Automàtica

Curs 18/19

Autor: Eduard Badia Claret

Director: Victor Barcons Xixons

Data: 5/07/2019

Localitat: Manresa

AGRAÏMENTS

Vull agrair el suport permanent de l'Helena, la meva persona preferida, que mai es cansa i sempre riu. També a la meva família, Francesc, Roser i Queralt, que en cadascuna de les meves decisions a la vida, sempre m'han donat el seu suport incondicional, el seu amor i la seva ajuda. També vull donar les gràcies al Victor, per l'ajuda que m'ha donat sempre que he tingut alguna dificultat o entrebanc durant aquest projecte. A en Jesús i al Manel per la seva col·laboració en la confecció i ideació de la maqueta i a la EPSEM en general, per l'etapa de formació de qualitat que amb aquest treball finalitza.

RESUM DEL PROJECTE

L'objectiu d'aquest projecte és el de proporcionar una plataforma experimental, que permeti l'avaluació de les afectacions sorgides al mantenir un eix rotatori levitant mitjançant el control magnètic. Aquest control es porta a terme amb un microcontrolador, electroimants i sensors de proximitat. Les afectacions poden ser tals com; la disminució de les vibracions que pateix l'eix, així com el seu fregament i desgast, o el consum elèctric al utilitzar o no els imants. A la maqueta s'hi ha afegit també sensors i reguladors, que permeten parametritzar i obtenir dades per posteriorment poder ser analitzades. El control de la maqueta i l'obtenció de les dades es gestiona a través d'IoT.

ABSTRACT

The purpose of this project is to provide an experimental platform that allows the evaluation of the affectations arising from the levitation of a rotatory shaft with magnetic control. This control is carried out with a microcontroller, electromagnets and proximity sensors. The affectations can be such as; the diminution of the vibrations that the axis suffers, as well as its rubbing and wear, or the electrical consumption when using the magnets or not. In the model has also been added sensors and regulators, which allows the parametrization and obtaining data to be subsequently analysed. The control of the model and the obtaining of the data is managed through IoT.

ÍNDEX

ÍNDEX.....	1
ÍNDEX DE FIGURES	3
1. INTRODUCCIÓ	5
1.1. OBJECTIU.....	5
1.2. FINALITAT	6
1.3. ABAST	7
2. ANTECEDENTS.....	9
2.1. SISTEMES PER MITIGAR LA VIBRACIÓ	9
2.2. LEVITACIÓ MAGNÈTICA	9
2.3. MANTENIMENT PREVENTIU	10
2.4. IOT I GESTIÓ REMOTA	10
3. DESENVOLUPAMENT	11
3.1. CONCEPTE.....	11
3.2. DESENVOLUPAMENT DEL PROTOTIP	12
3.3. MÈTODES DE CONTROL	16
3.4. HARDWARE	18
3.4.1. Convertidors DC-DC	19
3.4.2. Microcontrolador	20
3.4.3. Xips 74HC595 i 74HC165	22
3.4.4. Imants	23
3.4.5. Sensors de proximitat.....	24
3.4.6. Acceleròmetre	25
3.4.7. Sensor de corrent	26
3.4.8. Motor i el seu control (Pont H i Encoder)	27
3.4.9. Lector de targeta Micro SD	28
3.5. SOFTWARE	29
3.5.1. WiFi i Web	30

3.5.2. Imants i Sensors de proximitat	32
3.5.3. Acceleròmetre	33
3.5.4. Sensor de corrent.....	33
3.5.5. Lectura de l'encoder i control PID del Motor.....	35
3.5.6. Lector de targeta Micro SD	37
4. ESTUDI ECONÒMIC.....	41
5. IMPLICACIONS AMBIENTALS	43
6. CONCLUSIONS.....	45
6.1. RESULTATS OBTINGUTS.....	45
6.2. POSSIBLES MILLORES.....	45
6.3. VALORACIÓ PERSONAL	46
7. BIBLIOGRAFIA.....	47
8. ANNEXOS	49
8.1. ANNEX 1: ARTICLE DE REFERÈNCIA	49
8.2. ANNEX 2: ESQUEMÀTICS.....	59
8.3. ANNEX 3: CODI DE SOFTWARE	71

ÍNDIX DE FIGURES

FIGURA 1: PROTOTIP DE REFERÈNCIA.....	6
FIGURA 2: RESULTAT EXPERIMENTAL DEL PROTOTIP DE REFERÈNCIA	6
FIGURA 3: COIXINET DE RODAMENT	9
FIGURA 4: COIXINET AMB LUBRICANT.....	9
FIGURA 5: ESQUEMA DE LLAÇ TANCAT	11
FIGURA 6: CONCEPTE MAQUETA.....	11
FIGURA 7: PANTALLA PRINCIPAL DE LA WEB DE CONTROL DE LA MAQUETA	12
FIGURA 9: PANTALLA DIRECTORI DE LA WEB DE CONTROL DE LA MAQUETA	13
FIGURA 10: ELEMENTS MAQUETA.....	14
FIGURA 11: PLACA AMB PONT H	14
FIGURA 12: PLACA CONNEXIÓ A ENCODER	15
FIGURA 13: PLACA AMB SENSOR DE CORRENT.....	15
FIGURA 14: PLACA DE CONTROL (PROTOTIP).....	16
FIGURA 15: PLACA DE CONTROL (CIRCUIT IMPRÈS)	16
FIGURA 16: CONCEPTE PROJECTE DE REFERÈNCIA	17
FIGURA 17: DIAGRAMA DE CONTROL DEL PROJECTE DE REFERÈNCIA.....	17
FIGURA 18: DIAGRAMA DE CONTROL AMB QUATRE SENSORS	18
FIGURA 19: DIAGRAMA DE CONTROL AMB QUATRE SENSORS PER POLSOS.....	18
FIGURA 20: PLACA D'AVALUACIÓ DEL CONVERSOR DC-DC.....	19
FIGURA 21: ESQUEMÀTIC CONVERTIDORS DC-DC	20
FIGURA 22: PLACA D'EVALUACIÓ ESP32 DEKIT V1	20
FIGURA 23: ESQUEMÀTIC MICROCONTROLADOR	21
FIGURA 24: XIP 74HC595.....	22
FIGURA 25: ESQUEMÀTIC SHIFT REGISTERS.....	22
FIGURA 26: IMANT SG TYPE 58.....	23
FIGURA 27: ESQUEMÀTIC IMANTS.....	24
FIGURA 28: SENSOR DE PROXIMITAT CR30.....	24
FIGURA 29: ESQUEMÀTIC SENSORS DE PROXIMITAT.....	25
FIGURA 30: PLACA D'AVALUACIÓ DE L'ACCELEROMETRE MPU6050.....	25
FIGURA 31: ESQUEMÀTIC ACCELEROMETRE	26
FIGURA 32: CORVA CARACTERÍSTICA SENSOR DE CORRENT.....	26
FIGURA 33: ESQUEMÀTIC SENSOR DE CORRENT.....	27
FIGURA 34: MOTOR.....	27
FIGURA 35: ENCODER DEL MOTOR	27
FIGURA 36: ESQUEMÀTIC CONTROL MOTOR	28
FIGURA 37: PLACA D'AVALUACIÓ DEL LECTOR DE TARGETES MICRO SD	28
FIGURA 38: ESQUEMÀTIC LECTOR TARGETA MICRO SD	29
FIGURA 39: ENTORN DE DESENVOLUPAMENT DE CODI	30
FIGURA 40: DIAGRAMA DE FLUX, WIFI I WEBSERVER.....	31
FIGURA 41: DIAGRAMA DE FLUX IMANTS I SENSORS	32
FIGURA 42: MITIGACIÓ DEL SOROLL DE L'ADC GRÀCIES AL CONDENSADOR I MULTIPLES LECTURES	34
FIGURA 43: DIAGRAMA DE FLUX DEL LECTOR DE CORRENT	34
FIGURA 44: DIAGRAMA DE FLUX INPUT DEL PID.....	35

FIGURA 45: DIAGRAMA DE FLUX CONSIGNA PID.....	36
FIGURA 46: DIAGRAMA DE FLUX SORTIDA PID.....	37
FIGURA 48: DIAGRAMA DE FLUX GESTIÓ CARPETES.....	38
FIGURA 48: DIAGRAMA DE FLUX CREACIÓ FITXER.....	38
FIGURA 49: DIAGRAMA DE FLUX MÀQUINA D'ESTATS DEL FITXER	38
FIGURA 50: DIAGRAMA DE FLUX ESCRIPTURA FITXER.....	39
FIGURA 51: EXEMPLE FITXER RESULTANT	39

1. INTRODUCCIÓ

La vibració produïda degut al desequilibri de masses, que es produeix en sistemes mecànics rotatius, és un problema comú i que desemboca en pèrdues de diferents tipus, però que comporten, principalment, fregament i desgast. Aquestes pèrdues s'acaben traduïnt en mal funcionament, poca eficiència, costos energètics i econòmics, aquests derivats dels costos de manteniment o substitució dels elements del sistema.

L'ús de sistemes mecànics rotatius es troba molt estès en les nostres vides i especialment en el món industrial, per tant, trobar una bon sistema de control del desequilibri de masses que es produeix en aquests sistemes, suposa un estalvi important a nivell energètic i econòmic.

Algunes de les aplicacions que tenen aquests sistemes són:

- Turbines, emprades per l'obtenció d'energia elèctrica gràcies a l'energia hidràulica o tèrmica.
- Compressors, emprats per augmentar la pressió d'un fluid mentre en redueix el seu volum.
- Generadors, per tal de convertir un tipus d'energia (química, mecànica, lluminosa, ...) en energia elèctrica.
- Motors d'avió.
- Aerogeneradors.

Degut a l'alta presència que tenen aplicacions com les esmentades, ja existeixen actualment diferents sistemes per tal de mitigar l'efecte de les vibracions produïdes, aquests sistemes busquen doncs millorar el rendiment mecànic de les aplicacions, però també allargar-ne el seu temps de vida. Aquests sistemes solen incorporar transductors, actuadors i elements de control.

En els últims anys hi ha hagut varis estudis i aplicacions, per tal de disminuir aquests problemes intrínsecs dels sistemes rotatius, ja que eliminar completament el desequilibri de masses és impossible.

1.1. Objectiu

Partint de la idea i primer prototip (Veure Figura 1) desenvolupat per els professors Leonardo Acho, Francisco Palacios, Josep M. Rosell, Jesús Vicente Rodrigo i Víctor Barcons Xixons, i que s'explica en l'article publicat per ells mateixos "*A digital electromagnetic journal bearing control design to unbalance rotor vibrations reduction: An academic experimental realization*" (Veure Annex 1), en què es mostrava els efectes positius que tenia l'aplicació d'un control magnètic sobre un eix (Veure Figura 2), l'objectiu principal d'aquest projecte és el de desenvolupar una segona versió d'aquest prototip, per tal que permeti realitzar més proves i anàlisis amb diferents mètodes de control i condicions.



Figura 1: Prototip de referència

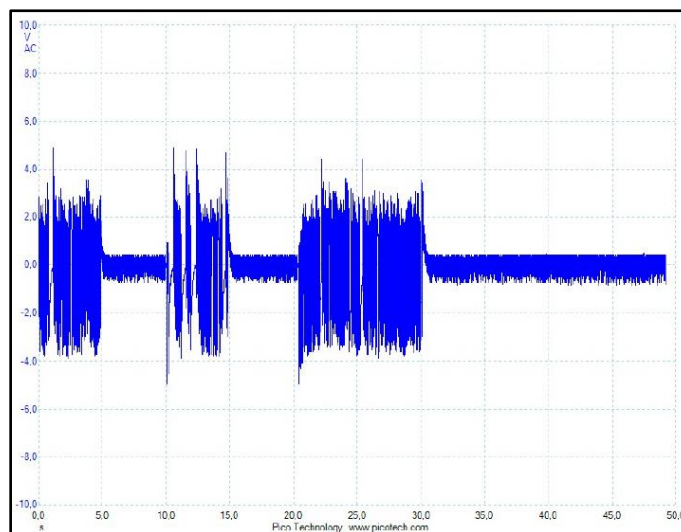


Figura 2: Resultat experimental del prototip de referència (Les zones amb menys amplitud de voltatge són aquelles en què el control magnètic estava activat)

1.2. Finalitat

La finalitat del projecte és doncs, la de proporcionar una eina de test i anàlisi que permeti realitzar diferents proves, amb diferents condicions, poder exportar-ne les dades resultants (en termes de vibració, consum energètic, ...) i que posteriorment podran ser analitzades.

Aquesta eina doncs, ha de permetre controlar tant el motor com els diferents actuadors en base a la informació rebuda dels sensors i dels paràmetres desitjats per l'usuari. Per tal de facilitar l'ús de la mateixa per part de l'usuari, aquest control es portarà de forma remota.

1.3. Abast

L'abast d'aquest projecte és doncs la confecció de la part de control de la maqueta, en que s'hi inclou la subseqüent part de hardware i el desenvolupament de software. Aquest control ha de permetre d'una manera fàcil, àgil i útil el control de la maqueta i l'extracció de dades obtingudes.

La confecció mecànica de la maqueta, queda fora de l'abast d'aquest projecte, així com també l'elecció dels imants, motor i sensors de proximitat instal·lats. Aquesta part ha estat realitzada per professors i tècnics de l'EPSEM.

També queda fora de l'abast d'aquest projecte els possibles testos a desenvolupar i el conseqüent anàlisi dels resultats obtinguts.

2. ANTECEDENTS

Per tal de realitzar aquest projecte, s'han tingut en compte diferents conceptes i en conseqüència els seus antecedents, per tal d'analitzar el marc teòric del projecte.

2.1. Sistemes per mitigar la vibració

El sistema més senzill que actualment hi ha per limitar la vibració d'un eix, és mitjançant un tub de diàmetre superior, en el que l'eix del motor s'hi situa dins. De forma que el moviment d'aquest queda limitat físicament per el tub que l'envolta. Per limitar encara més el moviment de l'eix a controlar, es situa entre un i l'altre un element lubricant (veure figura 4).

Un element més desenvolupat que aquesta proposta, és la del conegut com a coixinet de rodament (veure figura 3), aquests redueixen notablement els problemes de fricció.



Figura 4: Coixinet de rodament

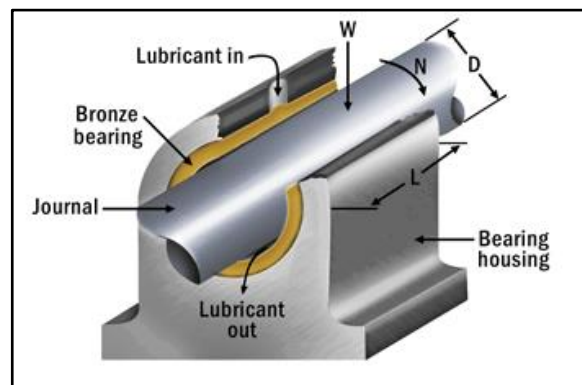


Figura 3: Coixinet amb lubricant

2.2. Levitació magnètica

La levitació consisteix en mantenir un objecte suspès en l'aire sense cap mena de contacte mecànic. Hi ha varies formes per què es produeixi la levitació d'un objecte, encara que no totes són vàlides per a la levitació d'un eix.

En aquest cas però, utilitzem la levitació magnètica, que és aquella que es produeix gràcies a l'acció d'un camp magnètic i, per tant, es contraposa a l'acció de la força de la gravetat. Hi ha múltiples aplicacions a la vida real que es basen en aquest concepte, tals com els rodaments magnètics o mitjans de transport.

La levitació magnètica es produeix gràcies a la repulsió existent entre dos pols d'imants, o bé mitjançant aquests per tal d'atreure o repel·lir d'altres objectes.

2.3. Manteniment preventiu

El manteniment preventiu és aquell que va destinat a la correcta conservació d'equips i sistemes, gràcies al seguiment i avaluació del seu funcionament, per tal d'aplicar mesures correctives de forma preventives, abans de que el sistema o alguna part d'ell deixi de funcionar o ho faci de forma no desitjada.

Hi ha varis tipus de manteniment preventiu:

- Programat: Es produeixen accions correctores/preventives de forma periòdica, ja sigui quan ha passat cert temps o cert nombre de cicles.
- Predictiu: Gràcies a les dades que pot proporcionar un sistema concret, en ocasions es pot predir quan el sistema pot començar a fallar o a tenir un mal funcionament. Cal però tenir un sistema que permeti obtenir informació i fer un anàlisi d'aquesta informació.
- D'oportunitat: Aprofitant períodes en que el sistema no serà emprat amb la freqüència habitual (com poden ser períodes de vacances dels treballadors), es fan accions preventives de manteniment per mantenir l'equip en les millors condicions possibles.

2.4. IoT i gestió remota

Des de fa una anys que se sent a parlar amb força de l'IoT, acrònim de "Internet of Things". Aquest concepte fa referència a la presència (cada cop més forta) de connexió a una xarxa inal·làmbrica de diversos dispositius que rodegen les nostres vides. Alguns d'aquests dispositius van des de rellotges fins a neveres, passant per cotxes o bicicletes entre molts d'altres.

L'IoT, permet que diferents dispositius connectats a la mateixa xarxa puguin intercanviar dades entre ells i per tant, poden comunicar-se. Es mitjançant aquesta comunicació que hi ha la possibilitat de controlar remotament un dispositiu amb un altre, que únicament necessitarà coneixen quin protocol segueix i per tant quines comandes cal enviar a través d'aquesta xarxa inal·làmbrica.

Avui en dia hi ha molts exemples de gestió remota, com per exemple:

- Programes de PC, que permeten l'ús d'altres PC's. Ja siguin d'una xarxa local o un connectat a través d'Internet.
- Drons o cotxes teledirigits, que es controlen mitjançant una aplicació de mòbil.
- Finestres i persianes, que es poden controlar amb una aplicació de mòbil o un control central.
- Equips de so, lavabos, cuines, sistemes de ventilació o d'il·luminació, neveres, etc.

3. DESENVOLUPAMENT

Per tal de desenvolupar i confeccionar la maqueta, s'ha partit de la maqueta i article de referència, abans ja esmentada, però s'hi ha incorporat elements més professionals/industrials i un enfoc més tecnològic amb la presència d'IoT per tal de fer el control de la maqueta. També, per poder analitzar quin és el sistema de control més eficient, hi ha dos tipus de control diferents als que trobàvem en la maqueta de referència.

3.1. Concepte

Gràcies a l'acció electromagnètica generada per quatre electroimants, es vol mantenir l'eix, sortint del motor, levitant. Com que el sistema és inestable i no lineal, és necessari aplicar un llaç de control tancat.

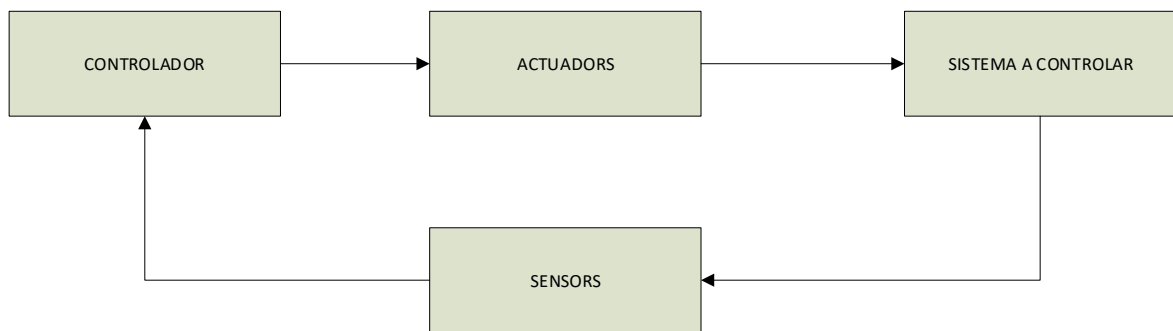


Figura 5: Esquema de llaç tancat

En aquest cas el controlador serà un microcontrolador, els actuadors (imants) ja esmentats tindran un efecte directe sobre el sistema a controlar (l'eix) i els sensors en llegiran contínuament l'estat de l'eix i proporcionaran aquesta informació al controlador, per tal d'obtenir un bon control del sistema.

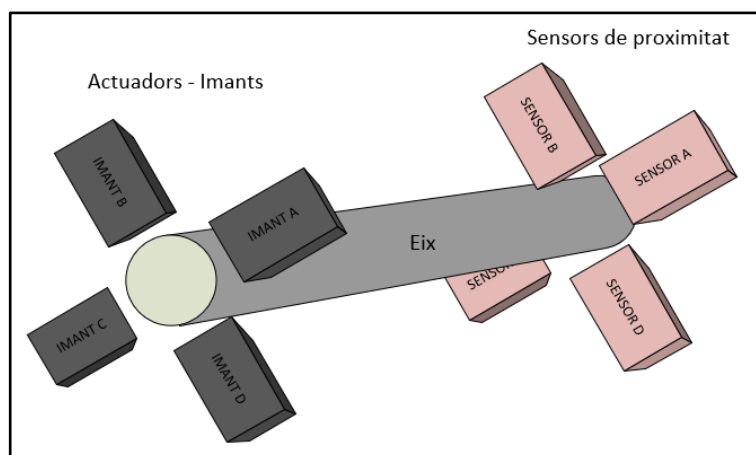


Figura 6: Concepte Maqueta

Així doncs, a mode resum, els sensors s'activen quan l'eix s'acosta a ells, aquesta informació arribarà al microcontrolador, que amb la lògica aplicada, gestionarà l'acció dels actuadors (imants) per tal de mantenir l'eix levitant, gràcies a l'acció de la força electromagnètica.

3.2. Desenvolupament del prototip

Aquesta maqueta ha estat confeccionada amb els elements fonamentals per l'objecte d'estudi, és a dir, amb el motor, eix, imants i sensors de proximitat, però també amb altres elements que permeten l'obtenció de dades interessants, com ho són les proporcionades per l'acceleròmetre (que ens proporciona dades de vibracions), o del sensor de corrent (que ens permet veure el consum energètic de la maqueta). També s'hi ha incorporat un lector de targetes Micro SD, per tal d'emmagatzemar aquestes dades i poder-les analitzar posteriorment.

Tots aquests elements es troben connectats a una placa d'avaluació del microcontrolador ESP32, un xip que està adquirint molta fama recentment ja que permet la connectivitat WiFi i BLE (Bluetooth Low Energy, no utilitzat en aquest cas). El microcontrolador és l'encarregat de gestionar tots aquests sensors i actuadors, però també és l'encarregat de generar una xarxa WiFi, que permet la connexió a una pàgina web (també generada pel microcontrolador), des d'on es pot controlar la maqueta (Veure Figura 7) i des d'on es poden descarregar i esborrar els fitxers generats (Veure Figura 8).

Control Maqueta Eix Levitador

	MOTOR	IMANTS	FITXER
Estat Actual:	OFF	OFF	StandBy
Canviar a:	ON	ON	START

Nom variable	Valor Actual	Unitat	Valor desitjat	Min % Max
Temps dels polsos	0	ms	<input type="text" value="0"/>	0 % 1000
PID: SetPoint	0	Rpm	<input type="text" value="0"/>	-1420 % 1420
PID : Kp	1.40	-	<input type="text" value="0"/>	>0
PID : Ki	1.00	-	<input type="text" value="0"/>	>0
PID : Kd	0.00	-	<input type="text" value="0"/>	>0

[Home](#)
[Directori](#)

EPSEM UPC 2019 - TFG Eduard Badia Claret

Figura 7: Pantalla principal de la web de control de la maqueta

Contingut carpeta "Maqueta"

Nom	Tipus	Mida
1	Dir	
--> 1.csv	File	267.235 KB
2	Dir	
3	Dir	
--> 1.csv	File	335.474 KB
4	Dir	
5	Dir	
--> 1.csv	File	1.496 MB
6	Dir	
--> 1.txt	File	325.378 KB
--> 2.txt	File	3.154 MB
7	Dir	
--> 1.txt	File	784.636 KB
8	Dir	

Escriu el nom del fitxer (amb l'extensió)

Exemples:
 - Per esborrar/descarregar fitxer dins d'una carpeta '/1/1.txt'
 - Per esborrar una carpeta (cal que estigui buida) '/1'

[Home](#)
[Directori](#)

EPSEM UPC 2019 - TFG Eduard Badia Claret

Figura 8: Pantalla Directori de la web de control de la maqueta

A nivell general doncs, a la maqueta hi podem trobar següents elements:

- 1) **Motor:** Que permet que l'eix giri.
- 2) **Junta eix motor:** Al ser quelcom flexible, provoca que l'eix tingui cert joc.
- 3) **Eix:** Tub de plàstic amb un element metàl·lic al seu extrem per tal de poder ser controlat pels imants, i un altre element a la zona dels sensors per tal de facilitar-ne la detecció.
- 4) **Imants:** Que han de fer possible la levitació de l'eix.
- 5) **Sensors de proximitat:** Que han de proporcionar la informació al microcontrolador sobre l'estat dels imants.
- 6) **Plaques electròniques:** On s'hi troben diferents plaques i components electrònics, que permeten al microcontrolador controlar i gestionar els diferents elements.

Aquests elements es troben distribuïts tal i com es pot veure en la figura 9.

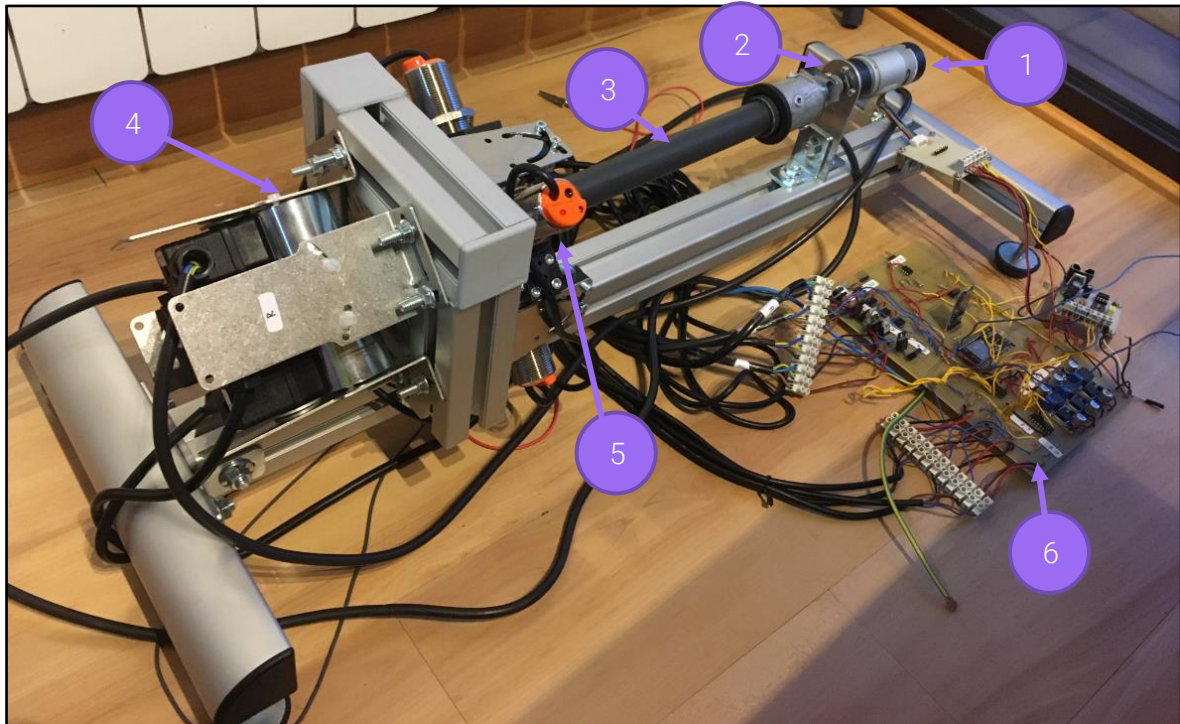


Figura 9: Elements Maqueta

Les plaques electròniques que es troben presents a la maqueta són les que s'expliquen a continuació, on les plaques 1, 2 i 3, del següent llistat, han sigut proporcionades pels professionals de l'EPSEM, mentre que la placa 4 (de control), és la que s'ha realitzat en aquest projecte:

- 1) **Placa amb Pont H:** Permet el control del motor, mitjançant una senyal PWM, que gestiona el voltatge que l'hi arriba al motor, i per tant, la potència amb que aquest actua. També ens permet fer girar l'eix en una direcció o una altra.



Figura 10: Placa amb PONT H

- 2) **Placa amb connexió a l'Encoder del motor:** Facilita la connexió de pins entre l'encoder instal·lat en el motor i la placa de control.



Figura 11: Placa Connexió a Encoder

- 3) **Placa amb sensor de corrent:** Incorpora el sensor de corrent i dos condensadors per al seu correcte funcionament.

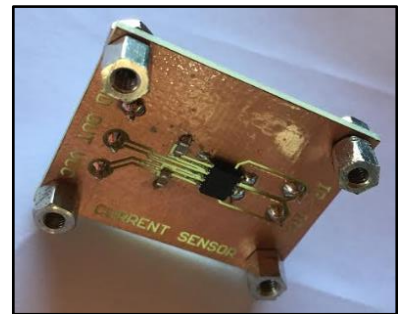


Figura 12: Placa amb sensor de corrent

- 4) **Placa de control:** Aquesta és la placa que s'ha realitzat, per tal de gestionar tots els elements que intervenen a la maqueta. És doncs la que ens permet controlar els actuadors (imants, motor), llegir els sensors (senyors de proximitat, de vibracions, de corrent) interaccionar amb els diferents elements (microcontrolador, lector de targeta Micro SD, les altres plaques de la maqueta).

Per tal de desenvolupar aquesta maqueta s'ha realitzat un primer prototip, mitjançant una placa de soldadura. Aquesta ha permès experimentar amb les connexions, components electrònics i les plaques d'avaluació, per tal d'arribar a una solució final en que tots els elements funcionessin correctament i s'entenguessin entre ells a nivell electrònic. El resultat d'aquest placa de prototip es poden veure en la figura 13.

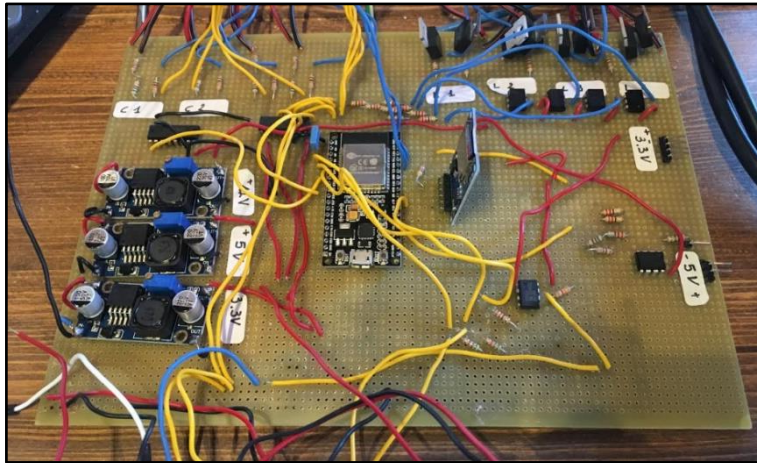


Figura 13: Placa de Control (prototip)

Un cop validada aquesta primera placa de control, se'n ha realitzat una de nova, que en aquest cas ha sigut una placa de circuit imprès, aquesta aporta més seguretat, ja que els cables no queden a l'aire, i per tant no es corre el risc de patir desconnexions o curtcircuits.

Aquesta placa s'ha dissenyat mitjançant el programa Fritzing, que també s'ha fet servir per realitzar els esquemes de la mateixa (Veure Annex 2). La placa resultant es pot veure en la figura 14.

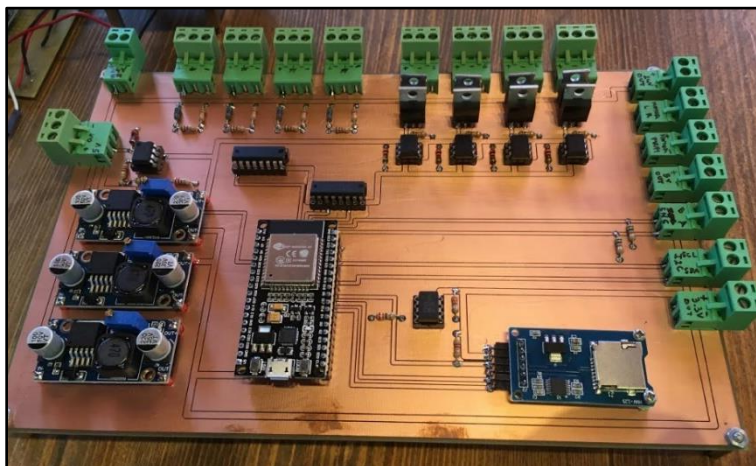


Figura 14: Placa de Control (Circuit Imprès)

3.3. Mètodes de control

A diferència del projecte de referència, aquesta maqueta pretén oferir a l'usuari dos sistemes de control dels imants. Per així poder estudiar i analitzar quin és el mètode més eficient per a la levitació de l'eix.

En el cas del projecte de referència, només hi havien dos sensors, un per a cada eix (Veure Figura 15). La informació de cada sensor servia per control dels dos imants del seu mateix

eix. Així doncs si el sensor detectava que l'eix estava pròxim a ell, el microcontrolador activava l'imant del costat oposat i desactivava el del seu costat. Mentre que si no el detectava, activava l'imant del seu costat per tal d'atraure'l cap a la zona on hi havia el sensor i desactivava el del costat oposat. Així doncs, sempre hi ha un imant actiu per a cada un dels eixos.

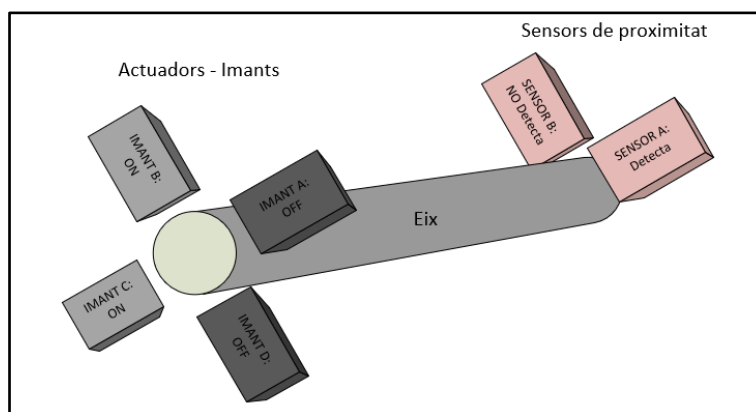


Figura 15: Concepte Projecte de referència

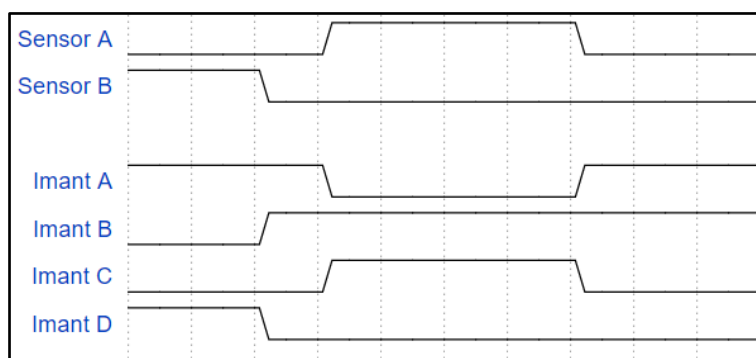


Figura 16: Diagrama de control del Projecte de referència

A diferència, en aquest projecte, s'hi han instal·lat quatre sensors (Veure Figura 6). De tal forma que cada imant depèn exclusivament del sensor del seu costat oposat. A part, el sistema ofereix dues formes de control.

La primera, de forma molt similar a la del projecte de referència, és la de mantenir activat l'imant contrari mentre el sensor està detectant. En aquest cas però, el microcontrolador, gestiona cada actuador en base al seu sensor contrari (Veure Figura 17). Així doncs el sensor A, controla l'imant C, el sensor B a l'imant D, el sensor C a l'imant A i el sensor D a l'imant B.

Aquest primer mètode de control es produeix, quan l'usuari ha escollit (des de la web) una durada de polsos de 0 milisegons.

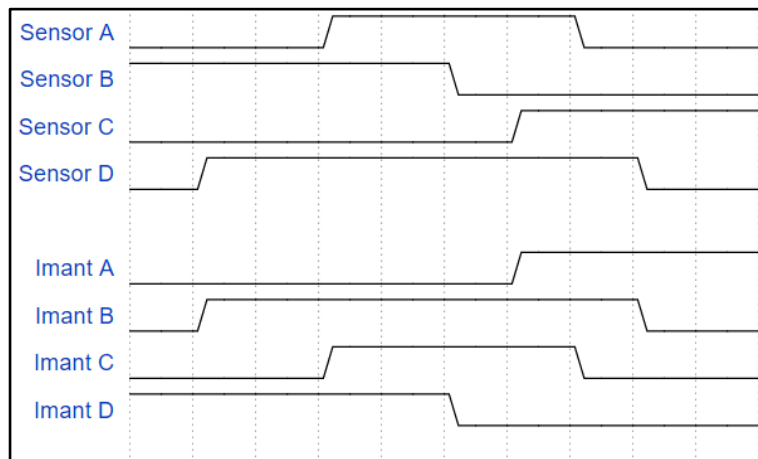


Figura 17: Diagrama de control amb quatre sensors

La segona, es tracta d'un control mitjançant polsos, de durada modificable per l'usuari. Quan es detecta que el sensor ha passat de no detectar a detectar (flanc de pujada), s'activa l'imant oposat durant els milisegons que hagi programat l'usuari (mitjançant la web). Aquests polsos poden tenir una durada de 1 a 1000 milisegons. La correspondència entre imants i sensors és igual que en el cas anterior.

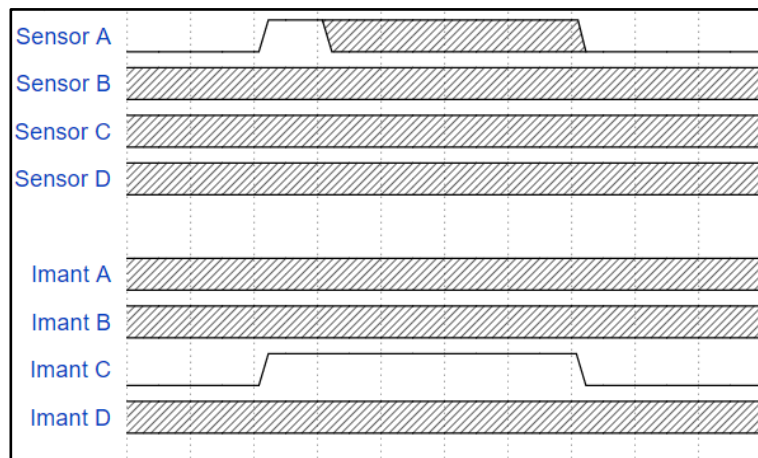


Figura 18: Diagrama de control amb quatre sensors per Polsos

3.4. Hardware

Al tractar-se d'una maqueta/prototip, s'han utilitzat, en alguns casos, plaques d'avaluació d'alguns xips, però per fer l'adaptació i tenir una bona comunicació entre elles ha sigut necessari l'ús d'altres xips, resistències, optoacobladors i amplificadors operacionals.

Per a fer funcionar de manera adequada la maqueta és necessari connectar-la a una font d'alimentació de 24V de tensió contínua. Aquesta tensió d'entrada serveix per alimentar:

- Els quatre imants.
- Els quatre sensors de proximitat.
- Els tres convertidors DC-DC.

3.4.1. Convertidors DC-DC

Aquests convertidors DC-DC són en forma de placa d'avaluació (Veure Figura 19) i del tipus reductor (Buck), serveixen per convertir un voltatge d'entrada a un de sortida que sempre tindrà menor tensió que la que hi havia a l'entrada. En el cas de les plaques d'avaluació emprades, accepten un voltatge d'entrada de 3 a 40V i un de sortida de 1'5 a 35V. La tensió de sortida d'aquests es regula mitjançant l'ús d'un potenciòmetre que va incorporat a la placa.

S'ha incorporat tres convertidors DC-DC a la maqueta , aquest converteixen el voltatge d'entrada de 24V a un voltatge de sortida de 12V, 5V i 3'3V.

Aquests convertidors serveixen per alimentar:

- 1) Convertidor de 24V DC a 12V DC.
 - o Els quatre optoacobladors dels imants.
 - o L'amplificador operacional del sensor de corrent.
- 2) Convertidor de 24V DC a 5V DC.
 - o El microcontrolador.
 - o L'encoder del motor.
 - o L'optoacoblador pel control PWM del motor.
 - o El sensor de corrent.
 - o El lector de targetes Micro SD.
- 3) Convertidor de 24V DC a 3'3V DC.
 - o Els xips 74HC595 i 74HC165
 - o L'acceleròmetre.

Aquest convertidor també ha servit per referència de pull-ups i entrada de senyal d'optoacobladors, com s'explicarà en la pertinent secció.

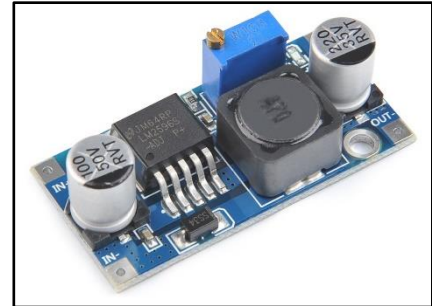


Figura 19: Placa d'avaluació del convertidor DC-DC

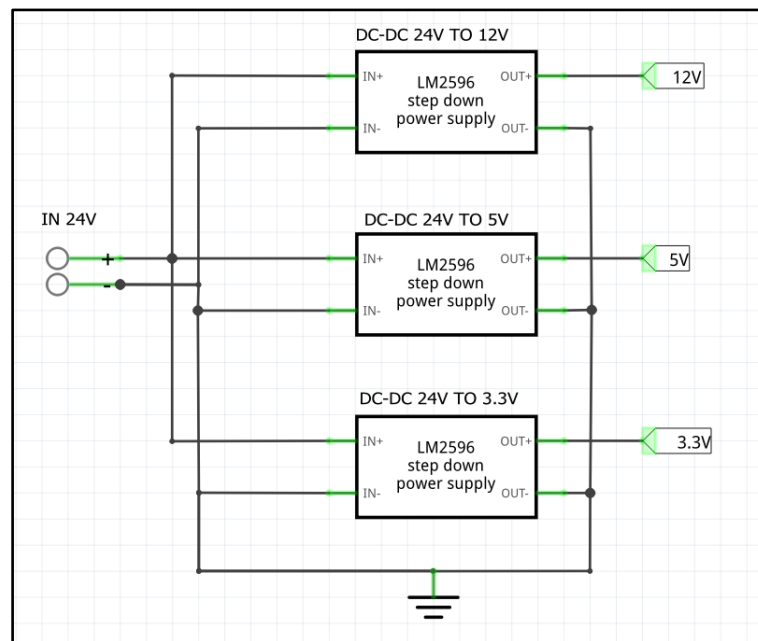


Figura 20: Esquemàtic Convertidors Dc-Dc

3.4.2. Microcontrolador

El microcontrolador escollit per gestionar totes les operacions que es duen a terme, és l'anomenat ESP-32 de la companyia Espressif. En aquest cas s'ha optat per utilitzar una placa de desenvolupament d'aquest microcontrolador (Veure Figura 21), per una qüestió de facilitat d'accés als pins i d'optimització del temps de dedicació al projecte.

Aquest microcontrolador, és famós avui en dia per el seu baix cost i la seva orientació cap al IoT, ja que integra WiFi i Bluetooth Low Energy, a banda d'un conjunt de pins que el converteixen en un microcontrolador molt versàtil, tot i que també amb algunes limitacions.

A nivell tècnic, destaca perquè:

- Inclou un processador Dual-Core de 32 bits, on un d'ells s'utilitza per la comunicació WiFi i l'altre per la resta de tasques.
- Disposa d'una memòria RAM de 520kB i una Flash interna de 448kB (pel Bootloader i funcions internes) i una externa de fins 16MB per al codi d'aplicació.
- Disposa d'un total de 32 pins d'entrada i sortida, tot i que algun d'ells no es poden utilitzar, com els 6 que gestionen la flash externa, o els 10 que formen part del ADC2, que només es poden utilitzar quan la WiFi no està activada. En quan a la funcionalitat dels pins, trobem que disposa de GPIO, PWM, convertidors ADC i DAC, es pot comunicar amb diferents protocols de comunicació com I2C, SPI, UART, ...
- Permet l'encriptació de codi a la flash.

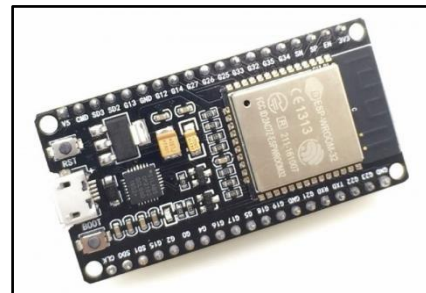


Figura 21: Placa d'avaluació ESP32 DEKIT V1

- Permet comunicació Bluetooth.
- Incorpora un coprocessador de molt baix consum.

Aquesta placa d'avaluació funciona amb un voltatge d'entrada de 5V, tot i que el microcontrolador funciona a 3.3V, però la placa ja porta incorporat un regulador per ajustar la tensió d'entrada de la placa a la que necessita el microcontrolador.

Com es pot veure en la figura 22, a la placa d'avaluació s'han connectat als diferents elements per a gestionar i controlar la maqueta.

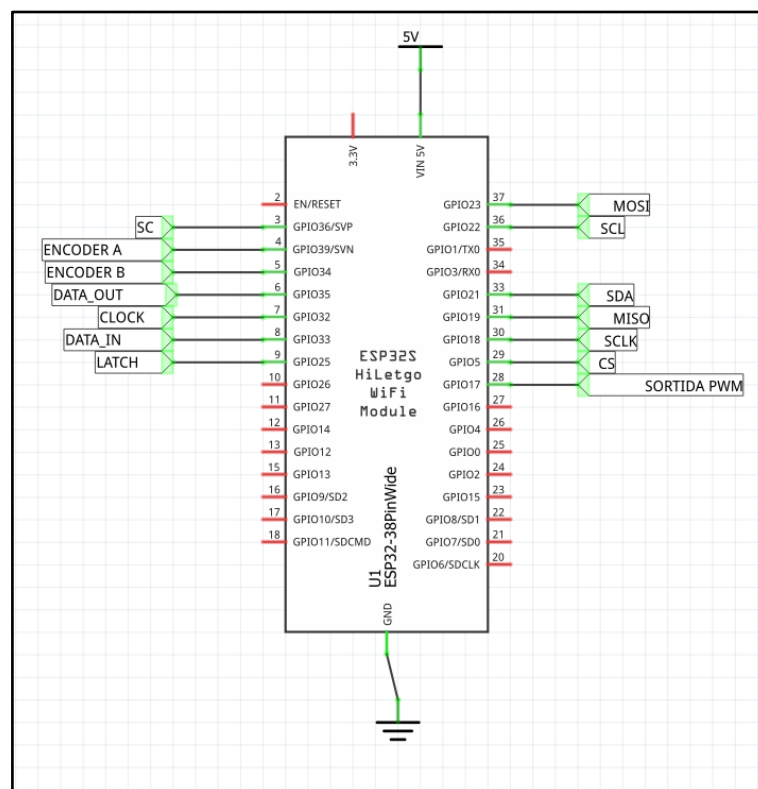


Figura 22: Esquemàtic Microcontrolador

3.4.3. Xips 74HC595 i 74HC165

Aquests dos xips s'han hagut d'incorporar a la maqueta degut a que el microcontrolador escollit no podia oferir funcionalitat a tots els pins quan el WiFi era activat. De tal forma que aquests xips han permès augmentar el nombre d'entrades (74HC165) i sortides (74HC595 (Veure Figura 23)) digitals de les que es disposaven.

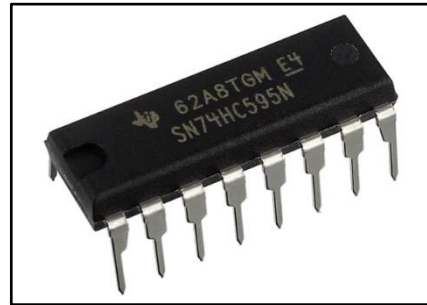


Figura 23: Xip 74HC595

Aquests xips actuen com a convertidors de dades; de Paral·lel a Sèrie (cas del 74HC165) i de Sèrie a Paral·lel (cas del 74HC595). De tal forma que amb la recepció d'un byte coneixem l'estat de 8 entrades digitals (en aquest cas se'n utilitzen només 4) i mitjançant l'enviament d'un altre byte, actuem sobre 8 sortides digitals (en aquest cas se'n utilitzen 6).

Així doncs, gràcies a aquests, es controlen 6 sortides, incloent l'activació dels imants i el sentit de gir del motor, i es llegeixen 4 entrades digitals provinents del senyal dels sensors de proximitat. Tot aquestes GPIO doncs es controlen amb tant sols 4 pins del microcontrolador:

- Una senyal de Clock comuna als dos xips.
- Una senyal de Latch comuna als dos xips.
- Una senyal d'entrada sèrie, provinent del 74HC165.
- Una senyal de sortida sèrie, que s'envia al 74HC595.

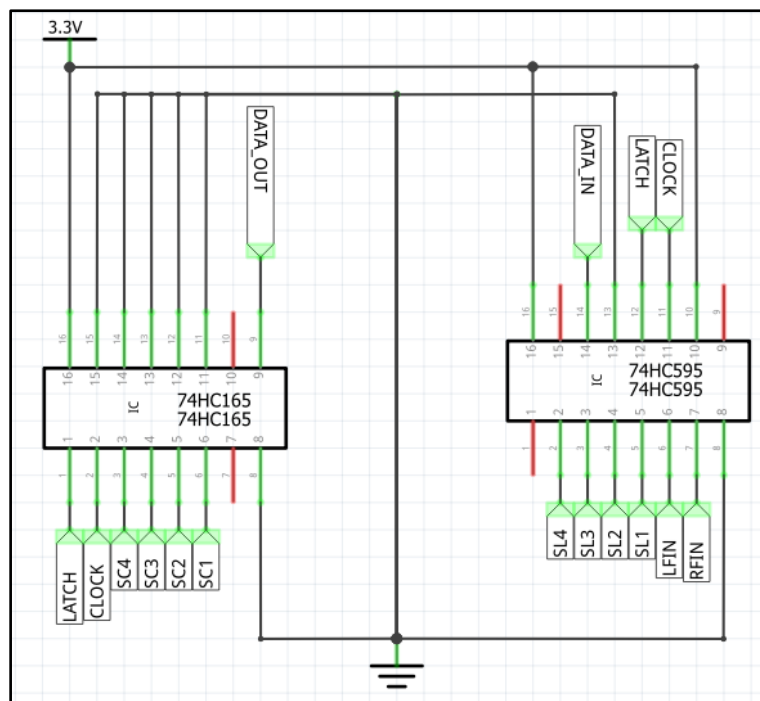


Figura 24: Esquemàtic Shift Registers

3.4.4. Imants



Figura 25: Imant SG Type 58

Els imants escollits per a realitzar la tasca d'atracció de l'eix, i així poder-lo mantenir levitant, són els SG Type 58 (Veure Figura 25). Per qüestions d'encabiment a la maqueta s'ha optat pels de tamany 50, que poden ser alimentats amb un voltatge de 12 a 24V DC (a la maqueta estàn alimentats a 24V DC). Són capaços de realitzar una força axial de 750N amb una potència de 5.4W. Per tant, amb un consumeixen aproximadament 0.225A quan són alimentats a 24V.

Per poder ser commutatats des del microcontrolador, mitjançant els xip 74HC595, abans explicat, ha sigut necessari incorporar diferents components electrònics per tal que els transistors que controlen aquests imants donessin valors de saturació (24V DC) o tall (0V DC), en funció de l'ordre del microcontrolador.

La senyal provinent del microcontrolador, així com la del 74HC595, és de tipus digital i commuta de 0V a 3.3V DC i per tal d'adaptar-la a l'amplitud que requeria l'imant s'ha optat per un Transistor d'efecte de camp (MOSFET) de canal baix (N). Aquest ens permet controlar una senyal d'alt voltatge amb una de baix voltatge. A aquest MOSFET se l'hi ha posat una resistència de PullDown, per mantenir la senyal a massa, sempre que no fos requerit el contrari pel microcontrolador.

Degut a que el MOSFET seleccionat (IRF540N), no era capaç de passar de tall a saturació amb els 3'3V, s'ha hagut d'incorporar un optoacoblador (4N35) entremig, per tal d'adaptar els 3'3V a 12V i en conseqüència una resistència limitadora, per tal de no fondre el diode LED del propi optoacoblador.

També comentar que s'ha incorporat un diode flyback o també anomenat antiparal·lel (BY229), ja que l'electroimant no deixa de ser una bobina i podria causar corrents contràries quan passi de ser activada a desactivada. D'aquesta forma evitem que l'electromagnetisme residual, no causi desperfectes a la resta del circuit. És per tant doncs una forma de protegir la resta del circuit dels possibles pics que pot generar la bobina al ser desactivada.

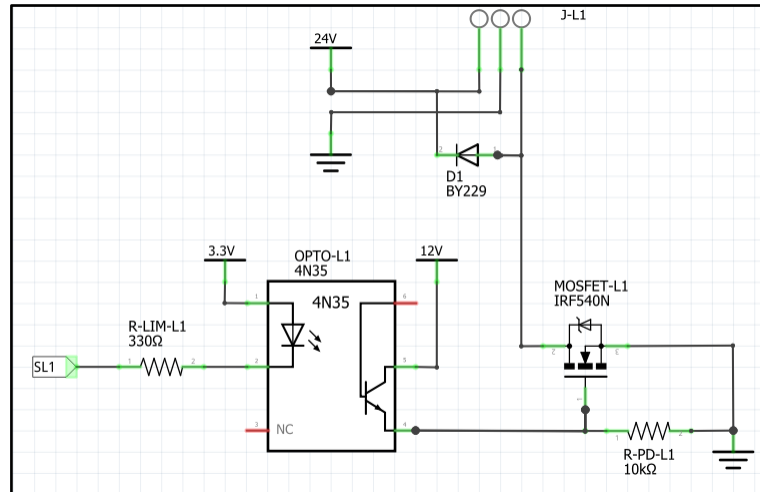


Figura 26: Esquemàtic Imants

3.4.5. Sensors de proximitat

Els sensors escollits són els CR30 de la marca Autonics (Veure Figura 27), aquests són de tipus capacitatiu i la seva inclusió a la maqueta s'explica per la necessitat de detectar en tot moment quin és l'estat de l'eix en termes de posicionament.

Aquests poden ser alimentats de 12 a 24V DC, en el cas de la maqueta ho estan a 24V DC. Cada un d'ells incorpora tres terminals, dos d'ells d'alimentació i un tercer de senyal. Aquesta senyal dona el voltatge d'entrada si el sensor està detectant un objecte dins el seu camp i 0V si no està detectant res. Aquesta senyal també va connectada a un LED del propi sensor, per tal de veure quin és l'estat de detecció en tot moment.



Figura 27: Sensor de proximitat CR30

Aquest sensor permet ser regulable, de tal forma que podem augmentar o disminuir el seu camp d'actuació. Aquest ajust es realitza gràcies a un ajustador de sensibilitat.

Ja que aquests sensors s'han alimentat a 24V DC la senyal de sortida commutarà entre 0V i 24V DC. Com que el microcontrolador i el xip 74HC165 treballen a 3.3V, aquesta senyal ha de ser ajustada a la mateixa amplitud. És per això que la senyal de sortida està connectada a un divisor de voltatge, que permet passar els 24V als 3.3V, tal i com es pot veure en la Figura 28.

Els valors de les resistències per realitzar el divisor de voltatge ha sigut calculada mitjançant la següent fórmula:

$$V_{out} = \frac{V_s * R_2}{R_1 + R_2}$$

Considerant que el voltatge d'entrada (V_s) és de 24 V i que volem una sortida (V_{out}) de 3.3V, primer s'ha fixat el valor d'una de les resistències i mitjançant la fórmula s'ha trobat l'altre valor necessari per tal de complir amb aquests valors.

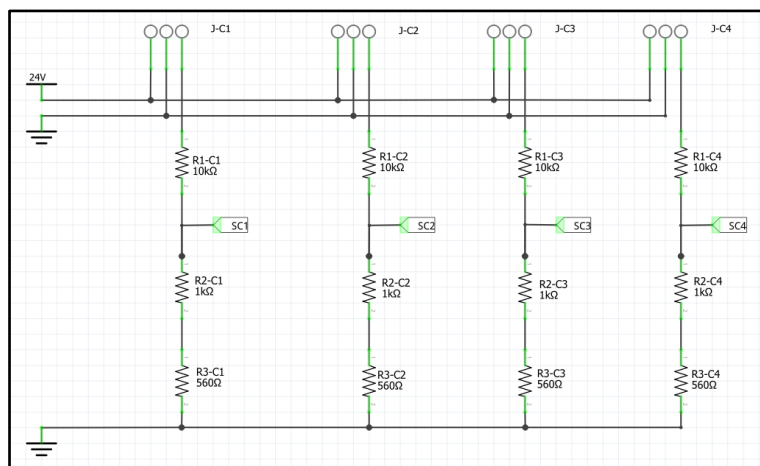


Figura 28: Esquemàtic Sensors de proximitat

3.4.6. Acceleròmetre

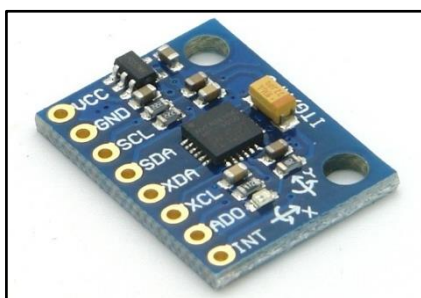


Figura 29: Placa d'avaluació de l'acceleròmetre MPU6050

Per tal de fer un anàlisi de l'efecte que tenen l'acció dels imants sobre l'eix i el motor, s'ha incorporat una placa d'avaluació de l'acceleròmetre MPU6050 (Veure Figura 29). Això s'ha fet amb la intenció de veure com canvien les vibracions que es produeixen quan els imants actuen i quan no. Aquesta placa s'ha situat a la zona on s'acobla el motor amb la junta que uneix a l'eix.

Es tracta d'un acceleròmetre i giroscopi, amb 6 graus de llibertat, i incorpora també un sensor de temperatura.

Aquest acceleròmetre és alimentat a 3'3V DC i es comunica amb el microcontrolador mitjançant el protocol I2C, que consta d'una línia de dades (SDA) i una de rellotge (SCL).

A nivell de hardware no ha estat necessari incloure cap element entre la placa d'avaluació del microcontrolador i la del acceleròmetre, ja que només farien falta resistències de pull-up, però en aquest cas es troben dins el microcontrolador i són activades per Software.

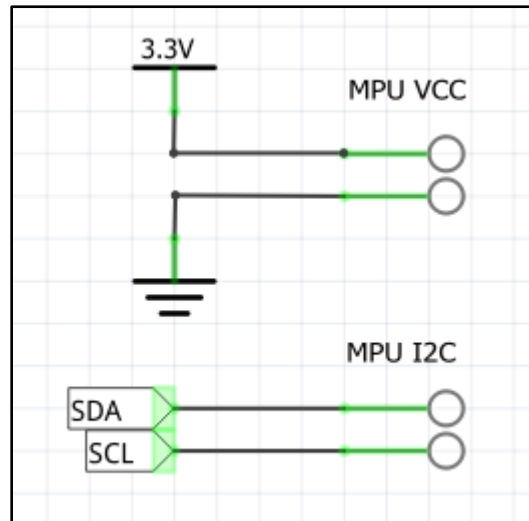


Figura 30: Esquemàtic Accelerometre

3.4.7. Sensor de corrent

Per tal d'avaluar els canvis de consum elèctric que suposen l'activació o no dels imants, així com la velocitat del motor, s'ha incorporat a l'entrada de la maqueta un sensor de corrent.

En aquest cas s'hi ha posat una placa prototip del xip ACS712T (Veure Figura 12), aquest accepta una intensitat màxima de 5A, la sensibilitat de sortida és de 185 mA / V i segueix un patró lineal, com es pot veure en la Figura 31, entre intensitat a l'entrada i voltatge de sortida.

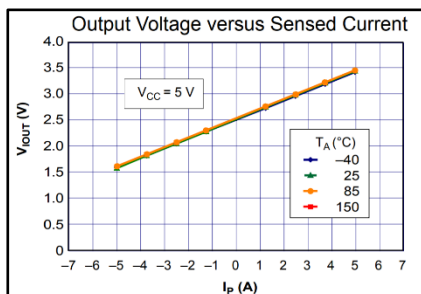


Figura 31: Corva Característica sensor de corrent

La lectura d'aquest sensor es fa mitjançant un dels ADC del microcontrolador, aquesta és però una de les parts més fluïdes de l'ESP32, ja que té un error diferencial de no linealitat de ± 7 LSB, i per tant només es poden garantir 9 bits monotònics.

Tot i això, i que s'han seguit les recomenacions del propi fabricant dels microcontroladors (Espressif), per tal d'optimitzar al màxim la lectura, aquesta no dona resultats del tot precisos.

Les mesures que s'han incorporat (a nivell de Hardware) per tal de tenir la millor lectura possible han sigut les d'adaptar el senyal per tal que vagi de 0 a 1V DC, això s'ha fet gràcies a un divisor de tensió, i la incorporació d'un condensador de desacoblament a l'entrada del microcontrolador, per tal de reduir al màxim el possible soroll.

A banda, ha sigut necessari incorporar un amplificador operacional configurat com a seguidor de tensió, per tal de que la senyal del sensor no caigués al connectar-la al microcontrolador, i una resistència de Pull-Down per a la mateixa senyal

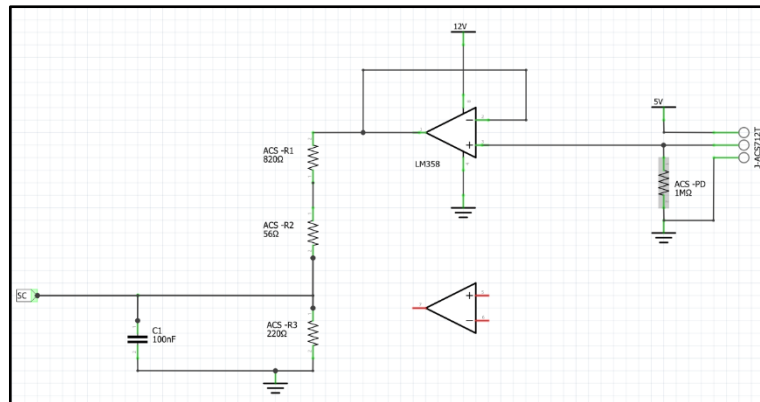


Figura 32: Esquemàtic Sensor de Corrent

3.4.8. Motor i el seu control (Pont H i Encoder)

El motor escollit per a fer girar l'eix de la maqueta és el Transmotec PD3557-24-5 (Veure Figura 33), aquest és un motor de corrent contínua, que pot anar alimentat fins a 24V DC, porta un reductor de 5:1 i una velocitat nominal (a 24V DC) de 990 rpm.



Figura 33: Motor

Per tal de controlar el funcionament del motor s'ha incorporat entre ell i el microcontrolador una placa prototip amb un Pont H (Veure Figura 10), aquesta s'alimenta a 24V DC. També s'ha utilitzat l'encoder que porta incorporat el motor (Veure Figura 34), que va alimentat a 5V DC.



Figura 34: Encoder del motor

Per controlar el motor, des del microcontrolador, s'utilitza una sortida PWM del mateix. La senyal va connectada a la placa del pont H i aquesta subvenciona més o menys voltatge al motor, en funció d'aquesta senyal PWM. Degut a que microcontrolador i Pont H funcionen a voltatges diferents (3'3V i 5V DC), s'hi ha hagut d'incorporar un optoacoblador d'alta freqüència (6n136) entre ells per tal d'adaptar-ne els senyals, conseqüentment s'ha hagut de posat una resistència limitadora per tal de no cremar el díode LED del seu interior i una resistència de Pull-up al col·lector del transistor BJT.

També, per poder llegir les senyals A i B del encoder, s'han hagut d'afegir resistències de Pull-Up, aquestes senyals van directes al microcontrolador. D'altre banda les senyals de control de direcció del motor (RFIN i LFIN) provenen del xip 74HC595.

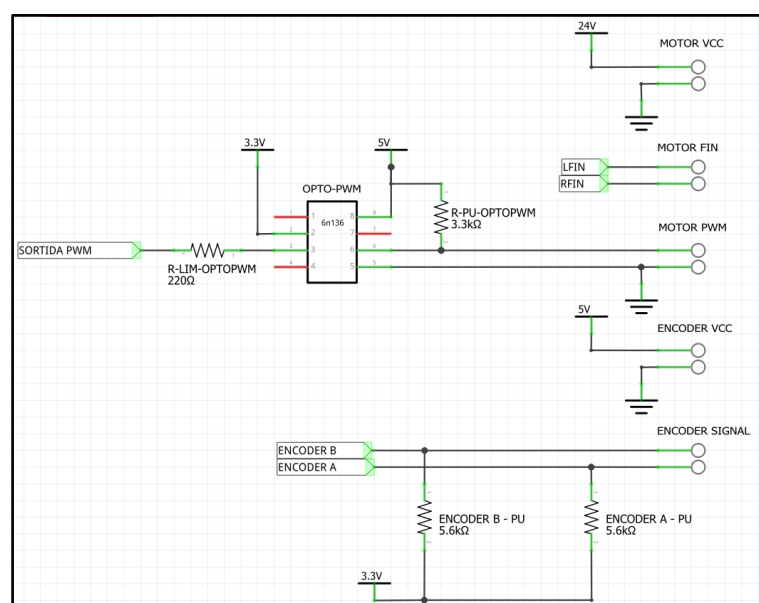


Figura 35: Esquemàtic Control Motor

3.4.9. Lector de targeta Micro SD

Per tal d'emmagatzemar la informació de l'acceleròmetre, el sensor de corrent i els paràmetres del motor, s'ha incorporat a la maqueta una placa d'avaluació d'un lector de targetes MicroSD (Veure Figura 36), ja que la capacitat del microcontrolador no permetia emmagatzemar la quantitat de dades desitjades. D'aquesta manera s'ofereix també la possibilitat d'extreure, les dades ja emmagatzemades, sense necessitat d'alimentar la maqueta (que ens permet descarregar fitxers mitjançant Wifi).

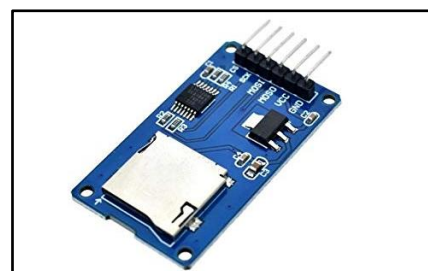


Figura 36: Placa d'avaluació del lector de targetes Micro SD

La placa en qüestió va alimentada a 5V DC i es comunica mitjançant el protocol de comunicació SPI, que consta de 4 línies:

- MISO: Senyal de dades (Master In Slave Out)

- MOSI: Senyal de dades (Master Out Slave In)
- SCLK: Senyal de Clock
- CS: Senyal de Chip Select

Per tal d'obtenir un correcte funcionament s'hi ha hagut d'incorporar una resistència de Pull-Up a la línia MISO.

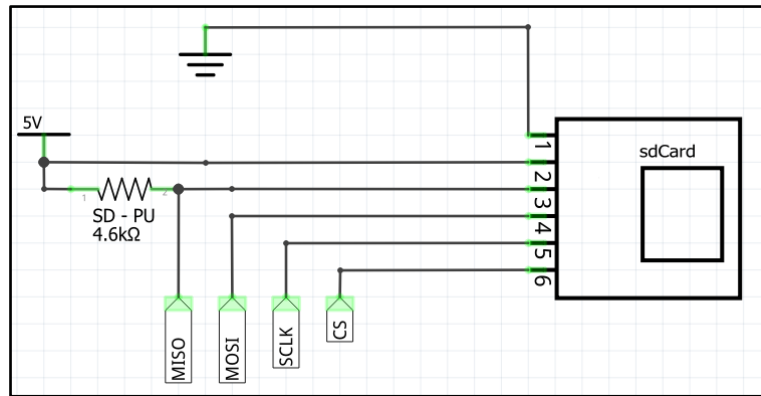


Figura 37: Esquemàtic Lector Targeta Micro SD

3.5. Software

Per tal de realitzar el Software de control de la maqueta, aquest s'ha desenvolupat amb l'ajuda del desenvolupador de codi d'Arduino (Veure Figura 38), ja que és una de les possibilitats que ofereix el mateix fabricant (Espressif).

Tot i ser un IDE senzill, que no permet l'ús dels breakpoints i que la forma més fàcil per depurar és l'ús del port Sèrie, si que té un punt molt favorable; la seva fama a nivell mundial i que conseqüentment comporta que hi hagi moltes llibreries i codis d'exemples a Internet. Aquest fet conjuntament a que el microcontrolador escollit també sigui, avui en dia, molt popular, han facilitat la confecció del Software a nivell de Drivers i llibreries.

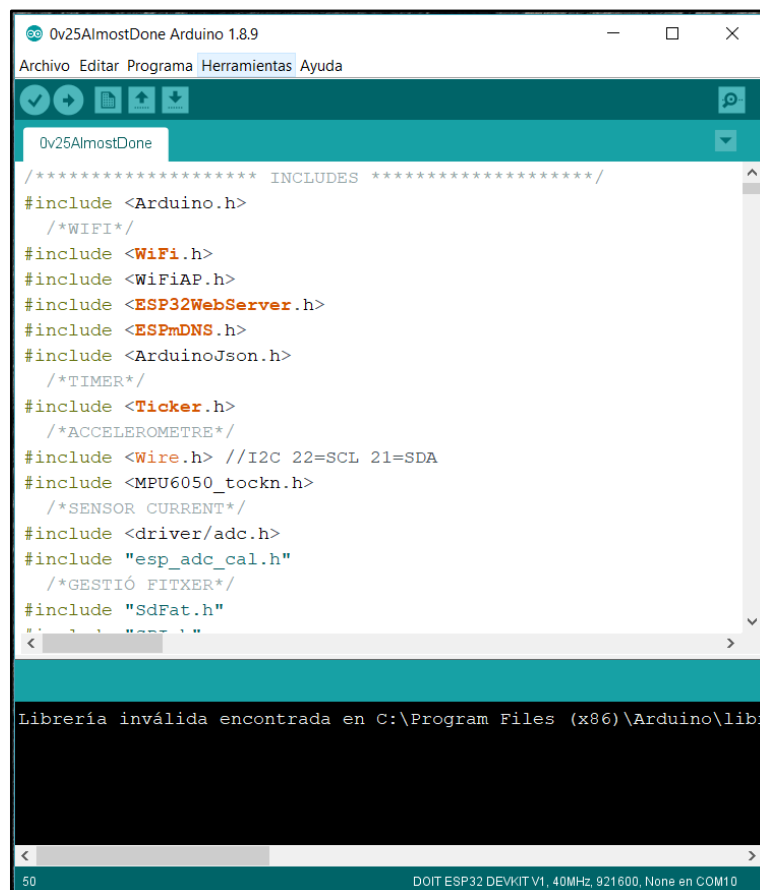


Figura 38: Entorn de desenvolupament de Codi

3.5.1. WiFi i Web

Amb la intenció d'oferir un entorn senzill i ràpid de poder controlar la maqueta i gestionar-ne posteriorment les dades obtingudes, s'ha optat per una solució sense cables, aquesta ens ho permet el microcontrolador gràcies a que permet la connectivitat Wifi i BLE, però per les característiques del projecte, només era factible via Wifi.

Al alimentar la maqueta, i per tant el microcontrolador, aquest crea una xarxa Wifi anomenada "maqueta" i amb contrasenya "abcd1234". Per tant l'ESP32 actua com a punt d'accés, però la xarxa que proporciona només permet la connexió d'un dispositiu a ella, per tal de protegir els elements de la maqueta i que no hi hagi conflictes entre dos usuaris.

Quan un dispositiu es connecta a aquesta xarxa, el microcontrolador actua també com a servidor web (amb IP: 192.168.14.14). Un cop l'usuari accedeix a aquest web, troba una pàgina que ha estat dissenyada bàsicament amb HTML i CSS (pel disseny de botons i taules), però també amb un petit script de Javascript (per enviar una cadena Json al servidor/microcontrolador).

Aquest web ens mostra quin és l'estat actual d'alguns paràmetres i opcions. En aquest cas, això és possible gràcies a la interacció de dades entre codi del microcontrolador (C i en conseqüència assembler) i el codi web. És en funció de l'estat de variables internes del

microcontrolador, que aquest envia unes línies o d'altres d'HTML quan actua com a webServer. Però també, en funció de les accions de l'usuari sobre la Web, aquesta modifica el valor de les variables del microcontrolador.

La pàgina, mitjançant peticions GET al servidor, permet modificar l'estat del motor, imants i fitxer, però també demanar la descarrega o esborrament de fitxers. D'altra banda mitjançant una petició POST, permet canviar els paràmetres de control del motor i el temps de durada dels polsos als imants, aquesta petició envia una cadena JSON al servidor, que és la que incorpora els elements i valors dels paràmetres.

La pàgina consta bàsicament de dues pantalles, una pel control de la maqueta (Veure Figura 7) i una altre per la gestió dels fitxers ja emmagatzemats (Veure Figura 8), tot i que aquesta acció també es pot fer extreien la targeta MicroSD del seu lector.

Per tal de incorporar totes aquestes funcionalitats en el Software, s'han utilitzat les llibreries: WiFi, WiFiAP, ESP32WebServer, ESPmDNS i ArduinoJson.

En el següent diagrama de blocs es pot veure quina és la lògica aplicada des de el microcontrolador per a realitzar les tasques de servidor web.

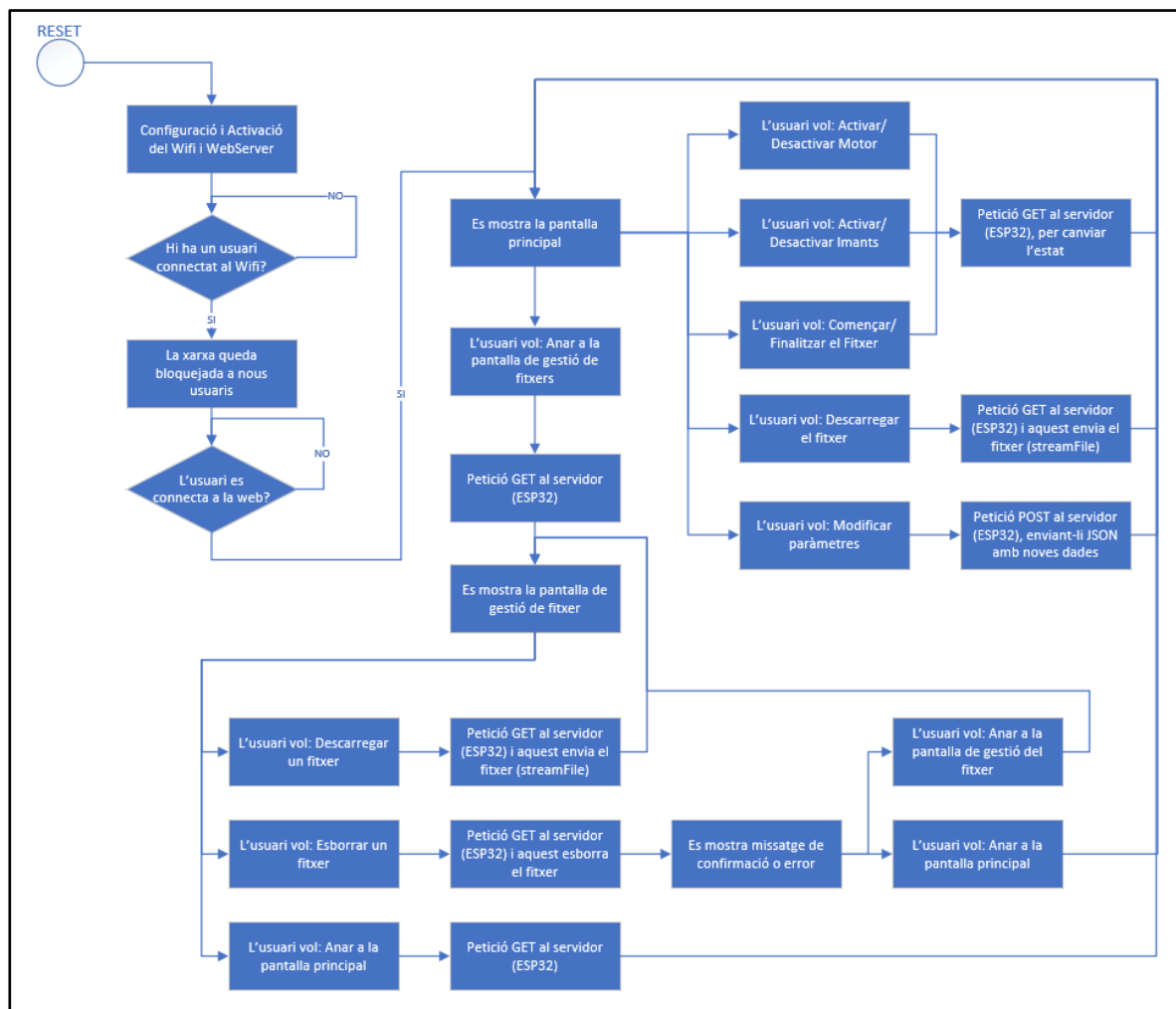


Figura 39: Diagrama de Flux, WiFi i WebServer

3.5.2. Imants i Sensors de proximitat

Com que el control i gestió d'aquests elements, s'ha fet mitjançant els xips 74HC595 i 74HC165, ha sigut necessari l'ús de les funcions `shiftOut` i `shiftIn` (que permeten l'escriptura i lectura de bytes en una GPIO), a banda de les de `digitalRead` i `digitalWrite` (que permeten la lectura i l'escriptura d'una GPIO), de fet, analitzant les primeres, s'ha observat que utilitzen les segones per arribar als seus propòsits. Totes aquestes funcions són pròpies de les llibreries d'Arduino.

Pel control d'aquests elements s'ha creat una funció anomenada `UpdateMagnets`, que es troba dins la interrupció de temps, que s'executa cada milisegon. En ella s'escriu quin ha de ser l'estat dels actuadors (mitjançant la funció `shiftOut`) en base a l'última lectura realitzada dels sensors (en l'anterior milisegon) i posteriorment es realitza la lectura dels sensors (mitjançant la funció `shiftIn`), abans de sortir de la interrupció.

Aquest procediment, el de primer escriure als actuadors i després llegir els sensors, és l'única forma viable de fer-se, ja que al només tenir 4 pins pel control dels dos xips (per les limitacions del microcontrolador al activar-se el Wifi), fa que hagin de compartir les senyals de clock i latch. En el següent diagrama de blocs es pot veure el funcionament d'aquesta funció.

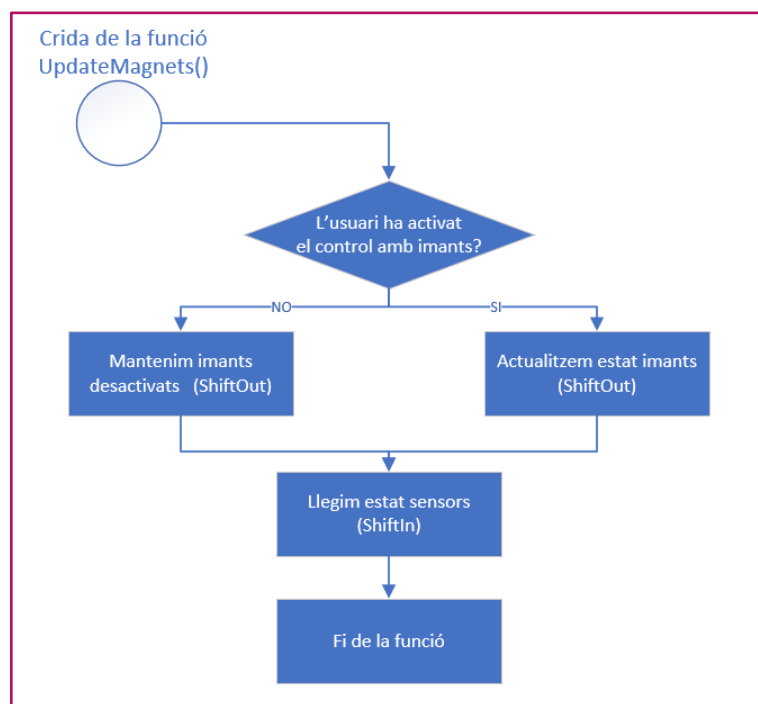


Figura 40: Diagrama de Flux Imants i Sensors

Com s'ha comentat anteriorment, hi ha dos mètodes de control dels imants (mitjançant polsos o no), la forma de canviar d'un a l'altre és modificant el paràmetre "Temps dels polsos", si aquest té un valor diferent de 0, els imants funcionaran mitjançant polsos de

durada determinada per l'usuari, mentre que si el valor és igual a 0, els imants actuaran fent cas del sensor oposat a ells.

3.5.3. Acceleròmetre

Per tal de llegir els valors que ens proporciona l'acceleròmetre, només ha calgut fer ús de dues llibreries; "Wire" per escriure mitjançant el protocol I2C, a través dels pins destinats a SDA i SCL del microcontrolador i "MPU6050_tockn" per interaccionar amb el dispositiu en qüestió.

En aquesta comunicació I2C, el microcontrolador actua com a Master i l'acceleròmetre com a esclau.

Aprofitant la llibreria d'aquest dispositiu, només ha calgut declarar una variable en forma d'estructura MPU6050, inicialitzar tant l'escriptura per I2C i l'acceleròmetre amb les funcions: Wire.begin i mpu6050.begin, i posteriorment en el bucle principal anem actualitzant els seus valors amb la funció mpu6050.Update. Per últim també s'han fet servir les funcions mpu6050.getXXX (pròpies de la llibreria del dispositiu) per tal d'agafar els diferents valors i posteriorment escriure'ls en el fitxer.

3.5.4. Sensor de corrent

Per a realitzar la lectura del valor que ens dona el sensor de corrent, aquesta informació arriba al microcontrolador en forma de senyal analògica. Per tant és necessari utilitzar un dels ADC (Analogic to Digital Converter) de l'ESP32, tot i que aquest, com s'ha comentat en l'apartat de Hardware, sigui un dels punts més dèbils d'aquest microcontrolador. Per tal de llegir aquest valor analògic s'ha fet ús de les funcions pròpies de la llibreria d'Espressif "esp_adc_cal" i la "driver/adc.h".

S'han configurat de tal manera que l'amplada de l'ADC sigui de 9 bits, sense atenuació (0 dB) i marcant com a valor de referència de l'ADC a 1128mV, valor que s'ha trobat experimentalment mitjançant les instruccions del fabricant, ja que cada xip d'aquest pot tenir un valor diferent en funció d'on i quan va ser fabricat.

Per llegir el valor d'aquesta entrada analògica, s'ha realitzat la funció ReadCurrent, en que es llegeix el valor de voltatge i en fa la conversió a Ampers. Aquesta funció es crida des del bucle principal. Aquesta realitza 100 lectures de l'ADC (també és una recomanació del fabricant, per tal de tenir el valor més fiable possible) per fer-ne posteriorment la mitjana i transformar el valor en volts a Ampers.

En la següent gràfica es pot observar les diferències d'incloure el condensador a l'entrada del sensor (explicat en l'apartat de Hardware) i les avantatges de fer múltiples lectures.

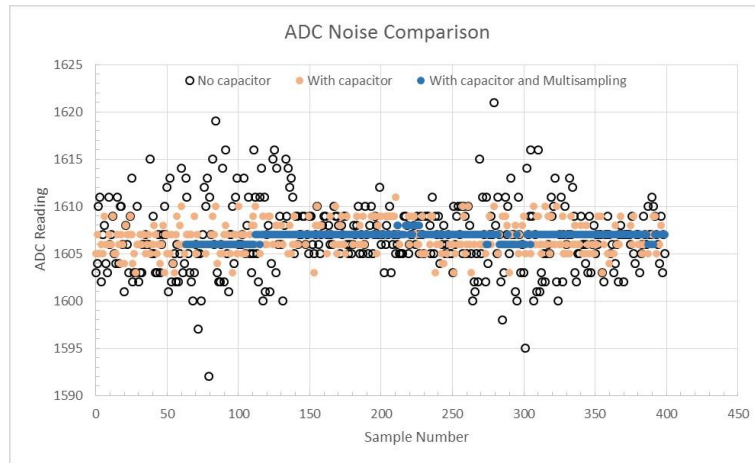


Figura 41: Mitigació del soroll de l'adc gràcies al condensador i múltiples lectures

L'explicació conceptual de la funció realitzada es pot veure en el següent diagrama de blocs.

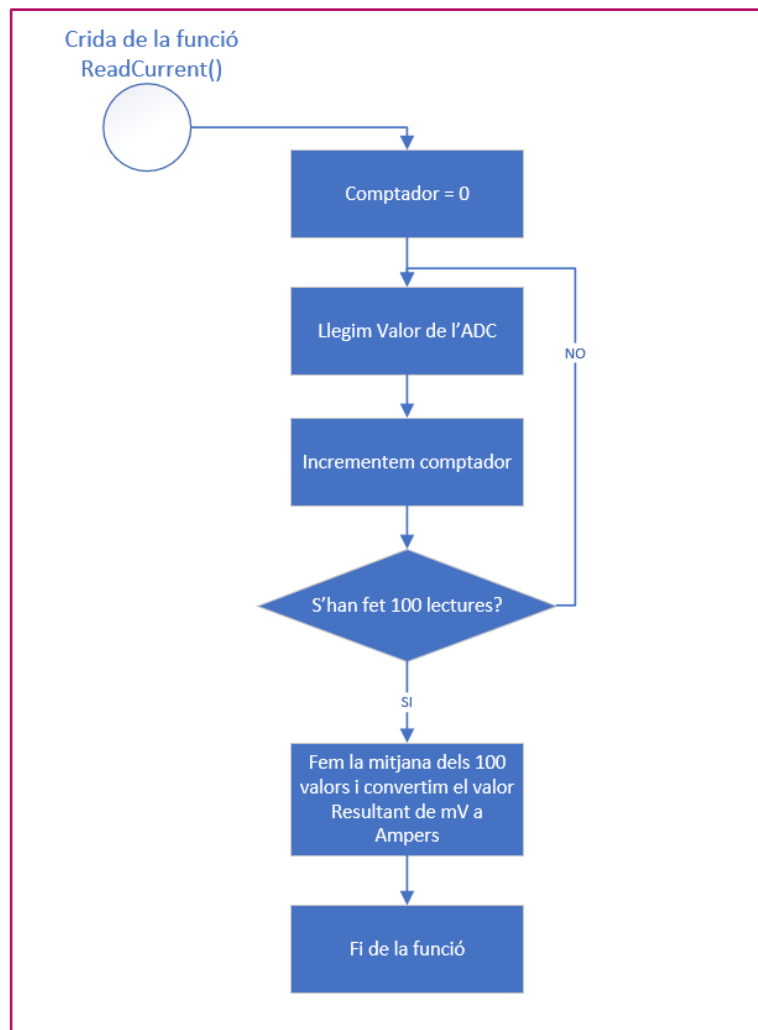


Figura 42: Diagrama de Flux del lector de corrent

3.5.5. Lectura de l'encoder i control PID del Motor

Per a realitzar el control del motor, ha sigut necessari incorporar un control PID en ell, això s'ha fet gràcies a la llibreria PID_v1. Els valors d'aquest controlador estan predefinits, però l'usuari els pot modificar (Kp: Constant Proporcional, Ki: Constant Integrativa, Kd: Constant Derivativa i Setpoint: Consigna) mitjançant el canvi de paràmetres que es pot fer des de la web.

L'entrada d'aquest controlador, s'adquireix gràcies a l'encoder del motor, que ens informa constantment de la velocitat amb que està girant. Per tal de llegir els valors de l'encoder, es fa mitjançant dues interrupcions (una per a cada canal de l'encoder), en que es compta el nombre de canvi de flanc per cada un d'ells i es compara amb l'estat de l'altre senyal. De tal forma que es realitza el càlcul de l'entrada en quadratura, per tenir una millor precisió de quin és el sentit i la velocitat de gir. Aquest càlcul es realitza cada 100 ms dins la funció motorCalc, que es crida dins la interrupció de 1ms.

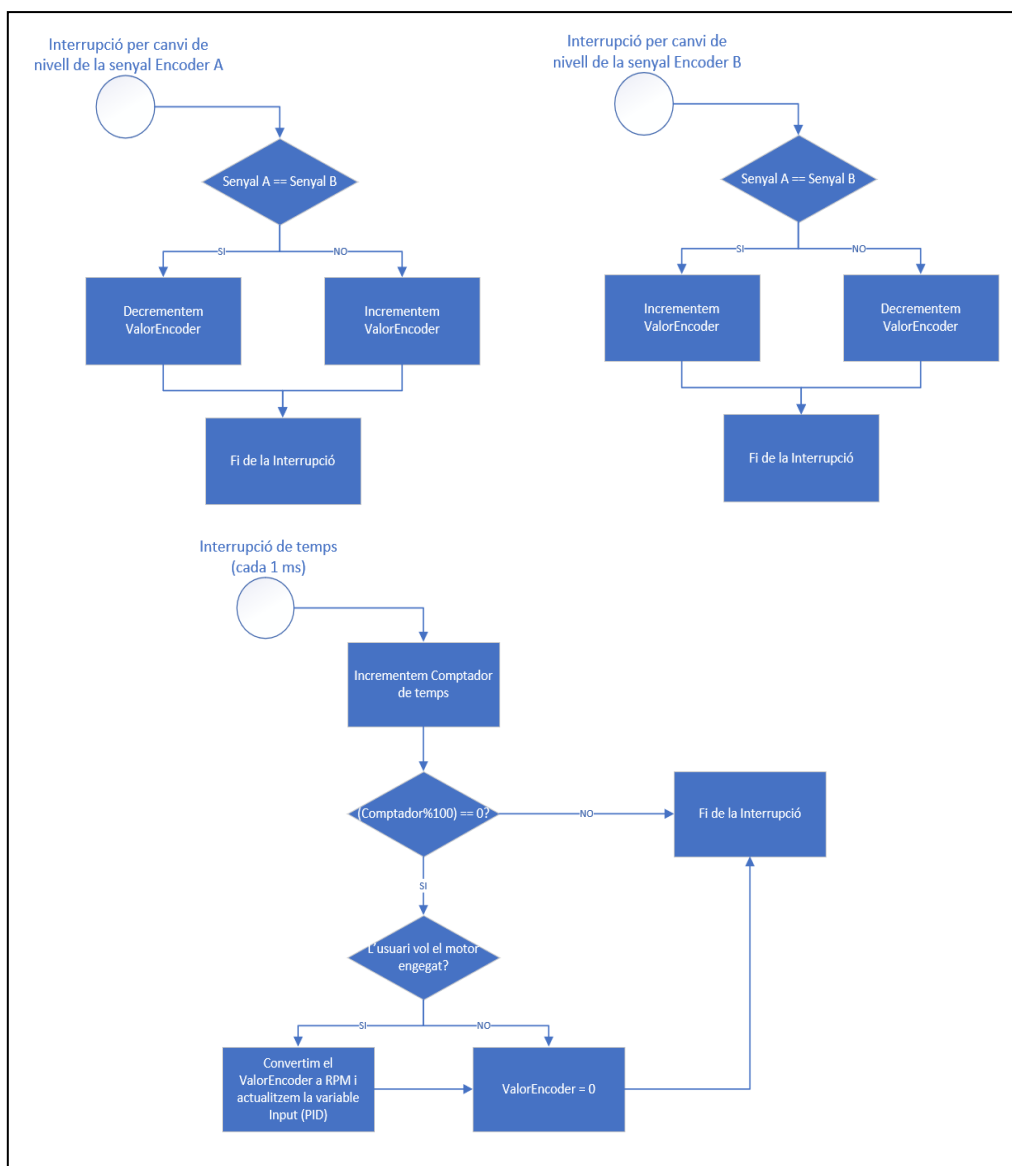


Figura 43: Diagrama de Flux Input del PID

Per tal de no realitzar canvis molt bruscos en el motor, quan l'usuari vol canviar de velocitat o de sentit de gir, s'ha afegit també la funció rampCalculator. Aquesta suavitza els canvis bruscos de velocitat sol·licitats, per tal de realitzar aquest canvis en forma de rampa i no d'esglaó. Per fer-ho possible, el programa conté dues variables per gestionar la consigna que s'introdueix en el PID. Una de treball (la que s'introdueix en el PID) i una de referència (la que escull l'usuari). Cada vegada que s'entra en la funció es comparen aquestes dues i es modifica la de treball per anar-la acostant cada vegada més a la de referència, fins que arriba a un punt que s'igualen.

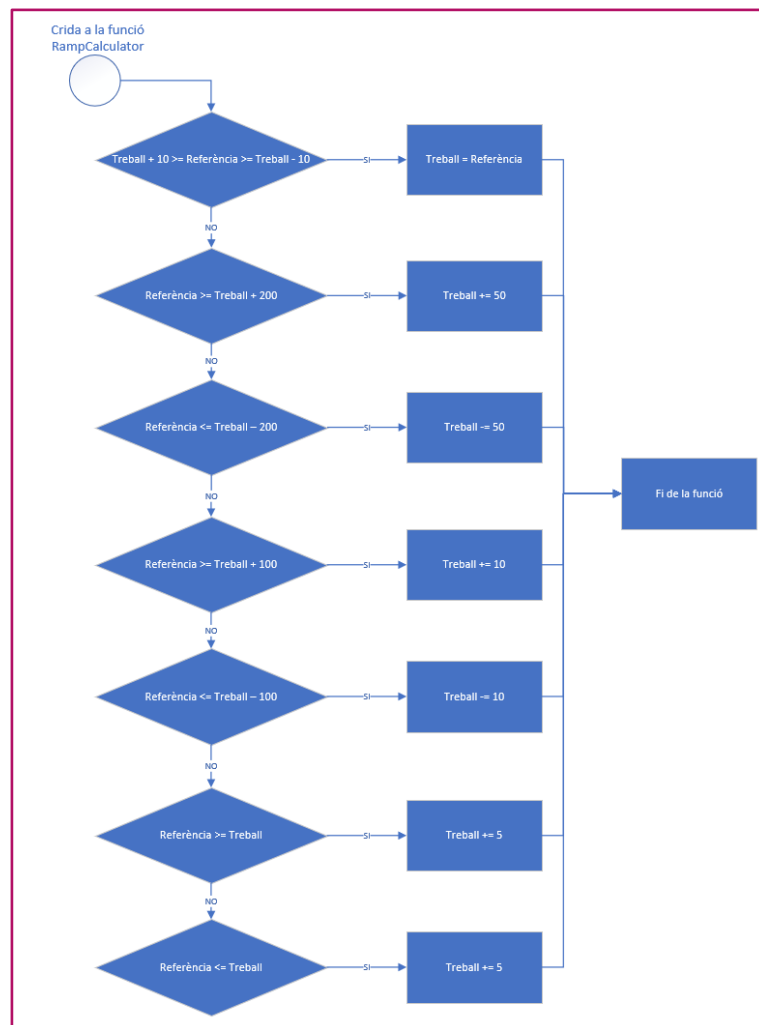


Figura 44: Diagrama de Flux Consigna PID

Per últim, per gestionar la sortida del PID, s'ha creat la funció UpdateMotor, que es crida en el bucle principal i s'encarrega; primer de actualitza el càlcul del PID i segon de treure la senyal PWM que controla la velocitat del motor i de gestionar dues sortides digitals (que s'han anomenat com RFIN i LFIN) del 74HC595 que permeten al microcontrolador gestionar el sentit de gir del motor:

RFIN	LFIN	Funció
0	0	L'eix gira "boig"
0	1	L'eix gira en sentit antihorari
1	0	L'eix gira en sentit horari
1	1	L'eix no gira

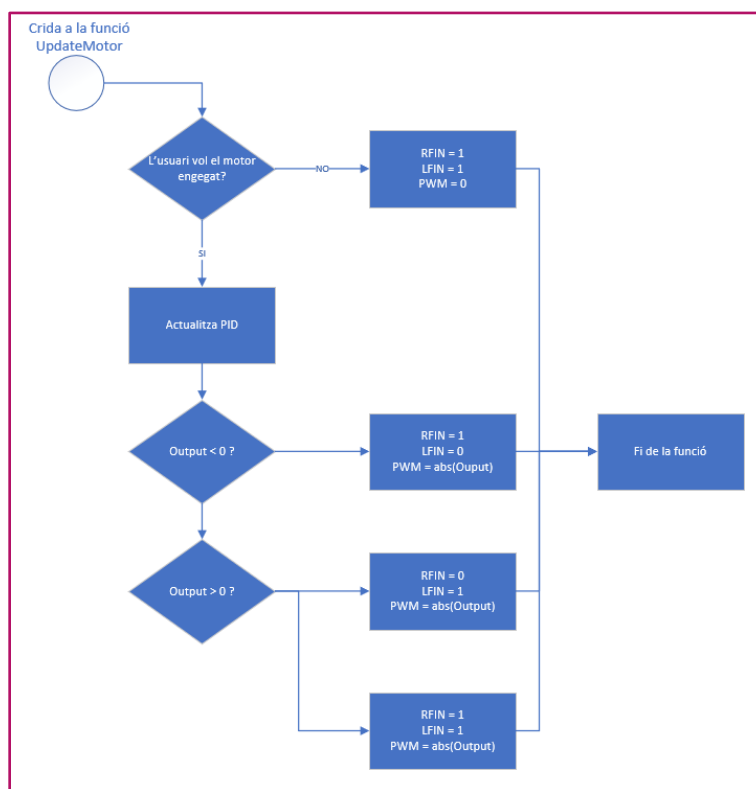


Figura 45: Diagrama de Flux Sortida PID

3.5.6. Lector de targeta Micro SD

La gestió de creació de la targeta MicroSd, consta de diferents parts i funcions.

Primerament, quan el microcontrolador arranca, aquest escaneja la carpeta "Maqueta" que hi ha guardada en l'arrel de la targeta (si no hi és, es crea automàticament) en busca de quina és l'última carpeta existent (que sempre serà un nombre enter) i es crea una nova carpeta, que tindrà com a nom el nombre consecutiu a l'última carpeta trobada. Així doncs, cada vegada que la maqueta s'engegui es crearà una carpeta, que tindrà com a nom un número. En cas de que es creï per primer cop la carpeta "Maqueta", també es crea a dins seu la carpeta "1".

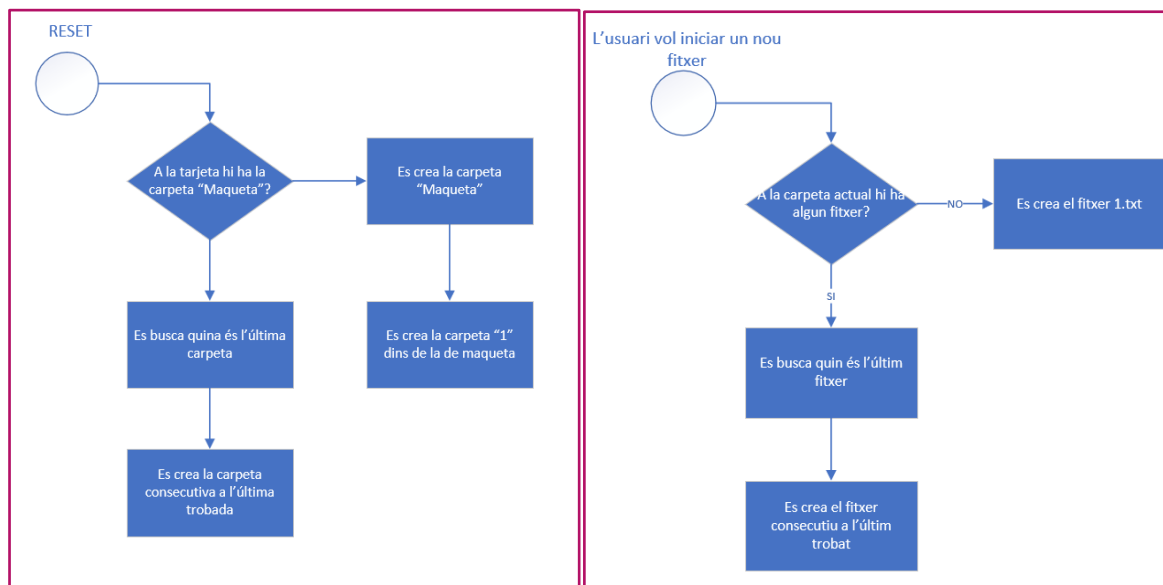


Figura 47: Diagrama de flux gestió carpetes Figura 47: Diagrama de flux Creació Fitxer

Quan l'usuari decideix que vol començar un fitxer, es fa primer una cerca dins la carpeta actual, per veure si ja existeix algun fitxer (que tindrà sempre un nom en forma de nombre enter), de tal forma que el nou fitxer tindrà com a nom el número consecutiu a l'últim fitxer trobat dins la carpeta. En cas de que sigui el primer, se l'hi assignarà el "1.txt".

Els fitxers s'escriuen amb l'extensió ".txt", ja que és el que genera arxius més lleugers, fet que facilita la seva escriptura i gestió (de descarrega i esborrament).

Un cop el fitxer s'ha creat, s'escriu una nova línia de dades, cada 5 ms, on s'hi afegeix la informació de l'estat actual de diferents valors obtinguts de l'acceleròmetre, el sensor de corrent i de l'estat del motor. Aquests se separen amb el símbol punt i coma ";", per tal de poder ser després tractats i separats de forma fàcil amb un gestor d'informació d'aquest tipus, com ho pot ser l'Excel.

A banda d'aquesta informació, cada vegada que l'usuari modifica paràmetres a través del web també queden registrats en aquest fitxer.

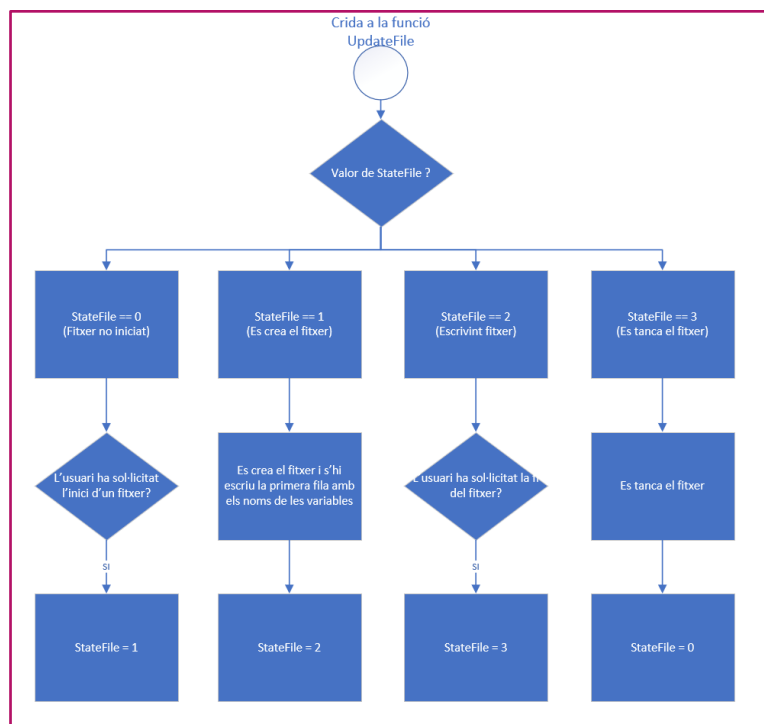


Figura 48: Diagrama de Flux Màquina d'estats del Fitxer

Quan l'usuari decideix finalitzar l'escriptura del fitxer, aquest es tanca i s'ofereix a l'usuari descarregar-lo (fet que també podrà fer sempre, així com també esborrar-lo, des de la pantalla "Directori" del mateix web).

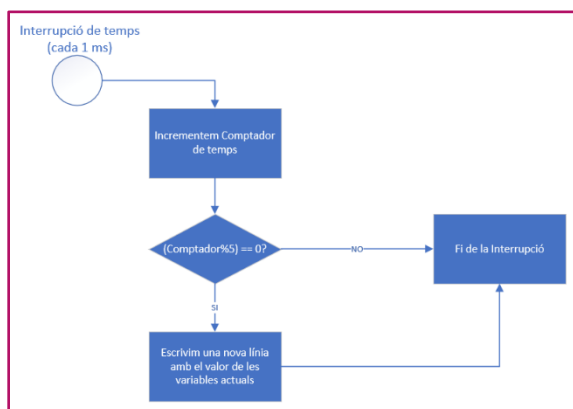


Figura 49: Diagrama de flux escriptura fitxer

Així doncs en la gestió del fitxer intervenen funcions que es troben a la part d'inicialització del microcontrolador, d'altres en el bucle principal i d'altres des de la interrupció de 1ms (encara que per una qüestió de rapidesa d'escriptura, només es fa cada 5 ms).

El fitxer resultant doncs consta d'una taula (Veure Figura 50) on les columnes són les variables i les files cada vegada que s'ha produït una escriptura. La idea de fer el fitxer així és per tal que sigui fàcilment interpretable i si fa falta dibuixable amb un programa com el Microsoft Excel.

	A	B	C	D	E	F	G	H	
1	millis	Output	Input	Setpoint	WorkSetpoint	Current	Temperatura	AccX	Acc
2	10	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.95	0.21
3	15	-85.12	60.72	0.52	0.52	-2.41	30.11	-0.95	0.21
4	20	-85.12	60.72	0.52	0.52	-2.41	30.11	-0.95	0.21
5	25	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.94	0.21
6	30	-85.12	60.72	0.52	0.52	-2.41	30.11	-0.95	0.21
7	35	-85.12	60.72	0.52	0.52	-2.41	30.01	-0.96	0.21
8	40	-85.12	60.72	0.52	0.52	-2.41	29.96	-0.95	0.21
9	45	-85.12	60.72	0.52	0.52	-2.41	29.96	-0.95	0.21
10	50	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.95	0.21
11	55	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.95	0.21
12	60	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.95	0.21
13	65	-85.12	60.72	0.52	0.52	-2.41	30.06	-0.95	0.21
14	70	-85.12	60.72	0.52	0.52	-2.41	30.01	-0.95	0.21
15	75	-85.12	60.72	0.52	0.52	-2.41	29.92	-0.95	0.21

Figura 50: Exemple Fitxer Resultant

4. ESTUDI ECONÒMIC

Al tractar-se d'un prototip, l'aplicació d'un estudi econòmic i de mercat no tenen sentit, però sí que en té el pressupost de la maqueta:

ITEM	MODEL	QUANTITAT	PREU/UT	COST
Motor	Transmotec PD3557-24-5-BFEC	1	135'80	135'80
Placa de desenvolupament amb microcontrolador	ESP32 NodeMCU Development Board	1	9'49€	9'49€
Convertidors Dc-Dc	QLOUNI LM2596 DC Buck Converter	3	1'99€	5'97€
Lector targetes Micro SD	Micro SD Storage Board TF Card Reader Memory Shield Module SPI for Arduino	1	1'89€	1'89€
Acceleròmetre	GY-521 MPU6050 Giroscope	1	4'49€	4'49€
Imants	58 - 0140 12 VDC - Electroimant, Tipo 58, 12VDC, 5.4W, 750N, IP51	4	69'75€	279'0€
Sensor de proximitat	CR30-15D	4	61'92€	247'68€
Targeta Micro SD	SanDisk Ultra Micro SD 32Gb	1	6'99€	6'99€
Amplificador Operacional	LM358N	1	0'36€	0'36€
Optoacobrador	4N35	4	0'43€	1'72€
Optoacobrador	6n136	1	1'05€	1'05€
Shift Register PISO	74HC165	1	0'38€	0'38€
Shift Register SIPO	74HC595	1	0'70€	0'70€
Díode	BY229	4	3'49€	13'96€
MOSFET Canal N	IRF540N	4	0'92€	3'68€
Resistències	Varis valors	32	0'5€	0'5€
Placa Pont H	BD6231	1	4'5€	4'5€
Placa Sensor de Corrent	ACS712T	1	4'6€	4'6€
COST TOTAL				722'76€

5. IMPLICACIONS AMBIENTALS

El principal motiu pel qual es vol veure l'impacte de mantenir un eix levitant mitjançant forces electromagnètiques és per veure quines diferències hi ha al tenir activat o no aquest control.

És lògic que si aconseguim fer levitar l'eix mitjançant el control amb imants, es reduiran les vibracions en el sistema, ja que passarem de tenir un eix amb molt de joc mecànic, a un eix molt més fix i estable. Per tant es reduiran les vibracions i estrès mecànic a l'eix i motor. Així doncs, tindrem una notable millora de rendiment i una pèrdua considerable d'aquestes vibracions, aquests beneficis es traslladaran directament en un allargament de la vida útil de la majoria d'elements que conformen el sistema, reduint així els costos de manteniment i allargant la vida útil dels mateixos. A més, gràcies a la possibilitat d'extreure dades de forma remota, podrem fer un control molt més exhaustiu i un seguiment de l'estat del diferents elements, que ens facilitarà les tasques de manteniment.

En contraposició a aquests components beneficiosos, trobem també un inconvenient important, a l'hora d'implementar aquest control amb forces electromagnètiques, que és el consum elèctric que això implica. Al tractar-se d'imants electromagnètics, aquests no deixen de ser bobines, les quals requereixen d'energia elèctrica per funcionar. En el cas de la maqueta, per exemple, els electroimants escollits tenen una potència de 5W quan treballen a 24V, per tant tenen un consum d'aproximadament 210mA cada un d'ells.

$$P = V * I \rightarrow 5W = 24V * 0'210A$$

Suposant que quan el control està activat, dos d'ells sempre estàn activats, això suposa un consum de 420mA, mentre que la resta de la maqueta (sense tenir activat el motor) consumeix aproximadament 100mA i quan el motor s'activa (al màxim de potència) la maqueta té un consum total aproximat d'uns 230mA. Per tant el consum augmenta considerablement quan passem a tenir el control de levitació actiu.

Per tant, si que podem dir que el control que s'aplica en la maqueta suposa uns beneficis a nivell de costos de manteniment i eficiència mecànica, però a la vegada també suposa un canvi negatiu en eficiència elèctrica. En resum doncs, serà en funció de l'aplicació que se l'hi vulgui donar a aquest sistema, que s'hauran d'avaluar els pros i contres esmentats, per extreure'n les conclusions pertinents i decidir si els beneficis superen les pèrdues o no.

6. CONCLUSIONS

6.1. Resultats obtinguts

Malgrat els esforços realitzats, la maqueta resultant d'aquest projecte no ha pogut fer possible la levitació de l'eix mitjançant els electroimants instal·lats en ella. La hipòtesis principal, d'aquest mal funcionament, és que es tracta d'un problema del sistema, en concret, tot apunta a que els sensors escollits en són els culpables.

Per arribar a aquesta hipòtesi s'ha comprovat que els imants són capaços de treure l'eix de qualsevol punt, tot i que això comporta col·locar-los molt junts, ja que no generen un camp magnètic prou gran com per atraure l'eix si aquest no es troba aproximadament a uns 5-7 mil·límetres de l'imant en qüestió.

També s'ha comprovat que el problema no vingués de la velocitat en què actualitzem els imants, que en el software es fa cada 1 mil·lisegon. Per comprovar això, s'ha fet la prova de posar un microcontrolador dedicat únicament a la lectura dels sensors i l'actualització dels imants, i que aquest gestionés aquestes tasques tant ràpid com pogués. De forma que la velocitat de refresc passava a ser molt més alta, ja que el microcontrolador es dedicava exclusivament a aquestes dues tasques, sense dependre de la interrupció de temps. Però els resultats eren els mateixos.

Aquestes proves, juntament a:

- La sensació, al treballar amb ells, que indiquen amb cert retard.
- Que incorporen electrònica dins seu i que es desconeix el seu funcionament intern (podria incloure histèresis o retards).
- La dificultat de calibrar-los tots amb el mateix ajust de sensibilitat.

Fan doncs que siguin els principals sospitosos de la impossibilitat de controlar magnèticament la levitació de l'eix del motor.

Tot i això el treball realitzat és fàcilment exportable si es canvien els sensors de proximitat o alguna altre part del sistema (eix, imants, motor, ...), ja que la placa dissenyada permet connectar nous elements a ella i desconnectar els actuals. Això sí, s'hauria de tenir present que la placa funciona amb voltatge de 24 Volts per als sensors i imants (que serien els elements a canviar per provar un nou sistema que controli l'eix amb èxit).

Per tant doncs, la maqueta entregada, es tracta d'una primera aproximació de l'objectiu, i aquest pot ser assolit si es canvia el sistema mecànic, sensors de proximitat i/o imants (ja que els actuals necessiten tenir molt a prop l'eix per a poder-lo atraure).

6.2. Possibles millores

Deixant de banda el sistema mecànic i sensors, abans comentats. Hi ha altres elements de la maqueta que podrien canviar-se o millorar-se per tal d'obtenir uns millors resultats o dades més fiables.

Un d'aquests components serien els imants, ja que els que es troben actualment a la maqueta, com ja s'ha comentat, necessiten tenir l'eix molt a prop per tal de poder atraure'l. A més, tot i tenir una força de 750N, aquesta és de forma axial i per tant si l'eix no es troba ben perpendicular a l'imant, l'atracció no és tal. A la vegada però, com més perpendicular es mantingui l'eix, més difícil serà pels sensors detectar un canvi de moviment. Per tant una possible millor seria canviar els imants per uns de més potents.

Un altre component a canviar podria ser el microcontrolador, tot i que això comportaria un replantejament molt gran del treball realitzat, per tal de trobar-ne un que no tingués les limitacions que mostra aquest. Aquestes han sigut principalment per la impossibilitat de fer servir molts dels seus pins quan s'activa el WiFi i la mala qualitat del seu convertidor analògic a digital. Una alternativa, per no canviar el microcontrolador, seria incorporar-ne un segon, i que ambdós es comuniquessin mitjançant per exemple Bluetooth o el port Sèrie. Aquest segon microcontrolador permetria:

- 1) No haver d'utilitzar els xips 74HC595 i 74HC165 ja que disposaríem de més pins. Per tant també es podria actualitzar més ràpid l'estat dels actuadors.
- 2) Si se'n incorporés un amb un millor convertidor analògic a digital, obtenir un valor molt més fiable de la lectura del sensor de corrent.

L'acceleròmetre incorporat funciona prou bé com a tal, tot i que no tant com a giroscopi. Per les funcions que es desitgen (per analitzar les vibracions), ja compleix perfectament, però si es volgués instal·lar en un altre lloc de la maqueta amb altres propòsits, aquest seria un altre element susceptible de canvi.

6.3. Valoració personal

Tot i no haver pogut fer funcionar amb èxit el control levitatiu que es pretenia, em sento molt satisfet per l'esforç i treball realitzat. Considero que la part que he confeccionat (la placa de control) i el software realitzat si que compleixen amb les perspectives inicials.

Considero que l'aprenentatge adquirit durant aquest grau a l'EPSEM, m'ha permès resoldre la majoria de conflictes que m'han anat sorgint durant el desenvolupament de la maqueta. Amb la realització d'aquest treball he pogut posar en pràctica molts dels coneixements adquirits, que m'han servit sobretot per dissenyar l'esquemàtic de components necessaris per fer funcionar tots els dispositius a incloure a la maqueta. Però també adquirir-ne de nous, com la programació amb el llenguatge C, HTML i CSS, el disseny d'una placa de circuit imprès, etc.

7. BIBLIOGRAFIA

ESPRESSIF. *ESP32 Overview* [en línia]. [Data de consulta: 25/02/19]. <<https://www.espressif.com/en/products/hardware/esp32/overview>>.

ESPRESSIF. *API Reference* [en línia]. Revisió 58df1d93. Shangai: 2019. [Data de consulta: 25/02/19]. <<https://docs.espressif.com/projects/esp-idf/en/latest/api-reference/>>

RandomNerdTutorials. *ESP32 Servo Motor Web Server with Arduino IDE* [en línia]. [Data de consulta: 13/03/19]. <<https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/>>

ARDUINO SM. *String indexOf() and lastIndexOf() Method* [en línia]. Revisió del 11/08/15. [Data de consulta: 27/04/19]. <<https://www.arduino.cc/en/Tutorial/StringIndexOf>>

W3Schools. *HTML: The Language for Building web pages* [en línia]. [Data de consulta: 23/03/19]. <<https://www.w3schools.com/>>

The Internet of Things with ESP32 [en línia]. [Data de consulta: 25/02/19]. <<http://esp32.net/>>.

TechTutoroalSX. *ESP8266 Webserver: Getting query parameters* [en línia]. [Data de consulta: 23/03/19]. <<https://techtutorialsx.com/2016/10/22/esp8266-webserver-getting-query-parameters/>>.

Dani No, Exploradores. *PRÁCTICA 15: Comunicaciones GET y POST* [en línia]. 11/05/18. [Data de consulta: 30/03/19]. <<http://www.esploradores.com/practica-15-comunicaciones-get-y-post/>>.

LastMinute Engineers. *How 74HC595 Shift Register Works & Interface it with Arduino* [en línia]. [Data de consulta: 04/05/19]. <<https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/>>.

Greiman. Github: *SdFat library* [en línia]. Greiman, 9/03/19. [Data de consulta: 05/04/19]. <<https://github.com/greiman/SdFat>>.

AVRFREAKS Forum. *Optocoupler qüestions/issues. And signal limits* [en línia]. Varis, des de 16/06/15. [Data de consulta: 05/04/19]. <<https://www.avrfreaks.net/forum/optocoupler-questionsissues-and-signal-limits>>.

Brett Beauregard. *Improving the Beginner's PID - Introduction* [en línia]. [Data de consulta: 25/05/19]. <<http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>>

Acho, Leonardo. *A digital electromagnetic journal bearing control design to unbalance rotor vibrations reduction: An acadèmic experimental realization. Energies*. 05/12/16.

8. ANNEXOS

8.1. Annex 1: Article de referència

A digital electromagnetic journal bearing control design to unbalance rotor vibrations reduction: An academic experimental realization

Leonardo Acho ^{1,†,*}, Francisco Palacios ^{1,2,†}, Josep M. Rossell ^{2,†}, Jesús Vicente Rodrigo ^{2,†}, and Víctor Barcons Xixons ^{2,†}

¹ Escola d'Enginyeria de Barcelona Est-EEBE, Universitat Politècnica de Catalunya-UPC, Spain; leonardo.acho@upc.edu

² Escola Politècnica Superior d'Enginyeria de Manresa-EPSEM, Universitat Politècnica de Catalunya-UPC

* Correspondence: leonardo.acho@upc.edu; Tel.: +34-934-137-240

† These authors contributed equally to this work.

Academic Editor: name

Version December 5, 2016 submitted to Energies; Typeset by L^AT_EX using class file mdpi.cls

Abstract: The main objective of this paper is to give an academic experimental platform design to evaluate magnetic control development for vibration reduction on a rotating shaft by using journal bearing and a cheap microcontroller device. In this microcontroller electronic integrated circuit, an simple digital controller is programmed by employing the assembler code. This code is the simplest one from the CPU point of view. According to our experiments, the propose digital controller is able to mitigate vibration in the rotating shaft. This experimentation may illustrates, for instance, a bearing control challenger to wind turbines power rotor generators, an important nowadays application and future technology development.

Keywords: Journal bearing; Rotating machine; Magnetic actuator; Vibration reduction; Digital control; microcontroller; Assembler code.

1. Introduction

It is well known that in rotating mechanical systems, vibration due to mass imbalance is a common problem to face with [1–4]. On these last 20 years, there have been an immense number of paper contributions on vibration control applied to rotating devices that may be classified into two main groups: 1) passive, and 2) active vibration control. On the other hand, the rotor and bearing system is one of the most important machine components numerously employed in many industrial rotating machinery such as turbines, compressors, generators, aircraft engines, and wind turbine [5], among others. Despite of the fact that the total elimination of the mass unbalance is actually impossible, it is crucial to control the vibration amplitudes within acceptable limits for a safe system operation [1–3,6]. Nowadays, in order to improve the machines performance, and hence to extend theirs working life, rotating elements incorporate transducers, actuators and control design with the objective to attenuate vibration amplitudes (or vibrations).

The journal bearing (or surface bearing) is the simplest type of bearing (rotor support) with no rolling elements (see Fig. 1). Therefore, the journal slides over the bearing surface. For instance, a simplest example of a journal bearing is a rotating shaft in a hollow. Hence they are compacts, lightweight, and they have a high load-carrying capacity, among other mechanical benefits such as friction reduction.

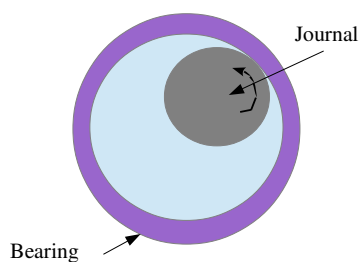


Figure 1. General representation of a journal bearing.

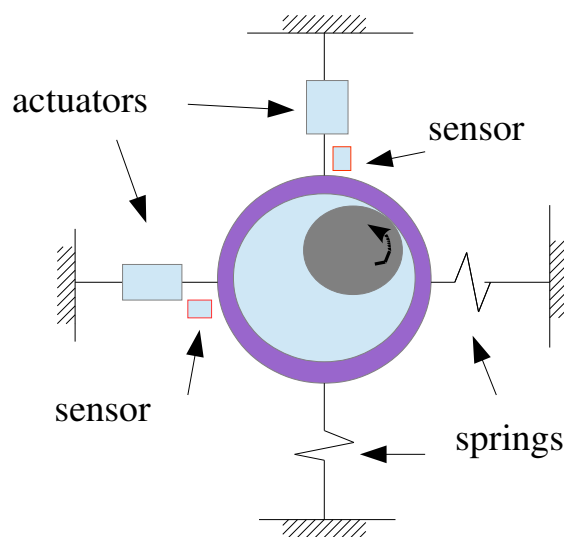


Figure 2. A journal bearing under vibration control: actuators and sensors.

To study possibilities on affecting rotor behavior for vibration control, the relative movement of the bearing bushing and the rotor is usually carried out, see Fig. 2. According to this figure, one vibration control option is to drive the bushing to modify the rotor dynamic. Therefore, some kind of sensors are required for position and/or velocity measurements, and actuators too. This stage problem has been extensively studied; see for instance [7,8,8–10] and references there in.

On the other hand, magnetic journal bearings are coming into increasing use in many recently industrial applications due to they are primarily employed to replace rolling elements, and they are a great potential option for vibration reduction on rotating machines [9,11,11,12].

Finally, Digital control offers several major benefits for journal bearing vibration control design in contrast to the analog control version. This because of the following facts: 1) the difficulty of on time changing control architecture with analog controllers; and 2) the sensitivity of the closed-loop to plant variation such as the fits of mechanical components [12]. Moreover, Digital control provides the ability to tune control parameters during installation tests, among other ones.

This paper discusses the digital design of a journal magnetic bearing control by using the micro-controller PIC12F508 by Microchip Inc. This controller is the cheapest one for control design,

and too easy to program [13]. In contrast to the control scheme given in Fig. 2; in our design, the bearing housing is fixed and the rotating shaft is magnetically manipulated to produce surface bearing and vibration reduction control, see Fig. 3 for a picture of our systems. We want to highlight that the digital control design given in this paper, including the prototype experiment, is just an academic system realization to illustrate an easy and self explanatory solution to the vibration control of magnetically manipulated rotating machine shafts by using low cost high digital technology (as the PIC controllers are [13]). Hence that this experiment is made to visually appreciate vibration control keeping its low cost for realization, and for teaching proposes.

The rest of the paper is structured as follows. Section 2 gives a description of our experimental platform for magnetic journal bearing for control vibration on a rotating shaft. Section 3 describes a digital algorithm development. An experimental result is shown too. According to the experiment data, our design has a good performance on rotor vibration reduction. Finally, Section 4 gives the conclusions.

2. Experimental platform design

To begin with our academic experimental platform design for a digital vibration control to a magnetic manipulated journal bearing, Fig. 3 shows a photograph of the from view of it. In this picture, four magnetic actuator coils are appreciated. Each coil is made by using 0.5 Kg of 0.8mm ϕ varnished wire. And the coils are wounded all around a C form ferromagnetic core. Fig. 4 shows the all part of this platform (Fig. 5 gives a dimensional perspective of it). On it we can see the rotor shaft (where the part inside the coils is made of iron). Also, two proximity sensors are appreciated. These sensors are from the Automatic firm of the CR-Series and AC-2 type with number part CR30-15A0. The PIC microcontroller and the digital shaft velocity sensor are illustrated too, among others basic instrumentation electronic designs. Moreover, we use an AC motor to drive (in open-loop control) our rotating shaft. This shaft is intentional imbalance to induce vibrations on it. Fig. 6 gives an schematic representation of our mechanism. Fig. 7 shows the power driver to each actuator coil to be digitally manipulated; that is, the input control to this driver comes from the PIC-12F508 microcontroller. Fig. 8 depicts the electronic instrumentation for each proximity sensor. Fig. 9 describes the interconnection signals to our PIC12F508 microcontroller from the instrumented sensors, and control signals to the previously explained power electronic drivers to each corresponding magnetic actuator coil. The used IC comparator was the popular LM293p. Fig. 10 gives the electronic system design for the shaft velocity sensor (an economical realization, [14]). This magnetic speed sensor uses 4 permanent magnets on the shaft as the sensing elements to capture the rotational angular speed of it. Finally, and for shake of explanation simplicity, the label names for the program given in Fig. 11 related to the pin names of the 12F508 pic-IC-microcontroller [13], are as follow:

- GPIO,04H: From the shaft vertical processed proximity sensor,
- GPIO,03H: From the shaft horizontal processed proximity sensor,
- GPIO,00H: to Coil 4 driver,
- GPIO,01H: to Coil 3 driver,
- GPIO,02H: to Coil 2 driver, and
- GPIO,05H: to Coil 1 driver.

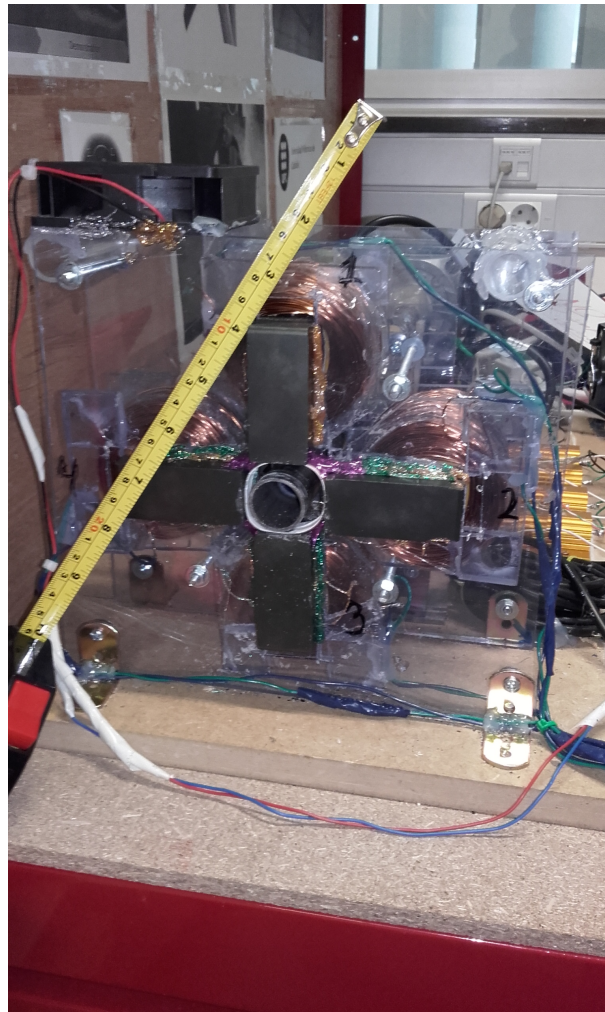


Figure 3. A close-up of our magnetic journal bearing mechanical design.

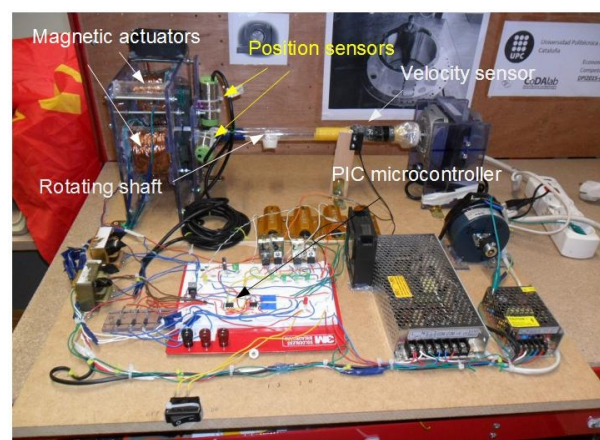


Figure 4. A photograph of the magnetic journal bearing mechanical design: an academic prototype for magnetic control evaluation.

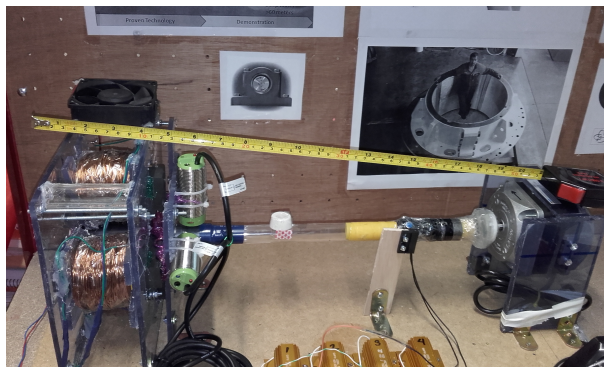


Figure 5. A dimensional perspective of the previous photograph.

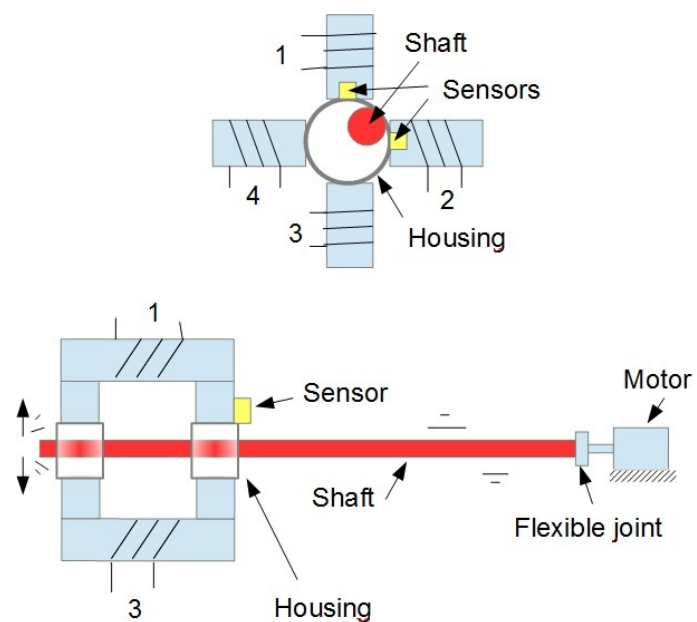


Figure 6. An schematic representation of our magnetic journal bearing system to vibration control; top: front view, and bottom: lateral view. In number are the magnetic actuator coils to identify them.

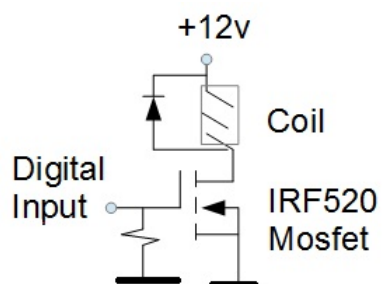


Figure 7. The power electronic driver for each magnetic actuator coil.

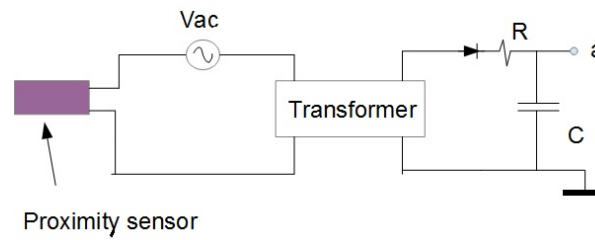


Figure 8. The block schematic diagram for the instrumentation design to each of the proximity sensors. They are two of these sensors: one for the shaft vertical movement and the other one for the shaft horizontal displacement. The point *a* goes to its corresponding comparators described in Fig. 9. In this diagram, $R = 330\Omega$ and $C = 47\mu F$, and the transformer is the reduction type one from 220Vac to 15Vac.

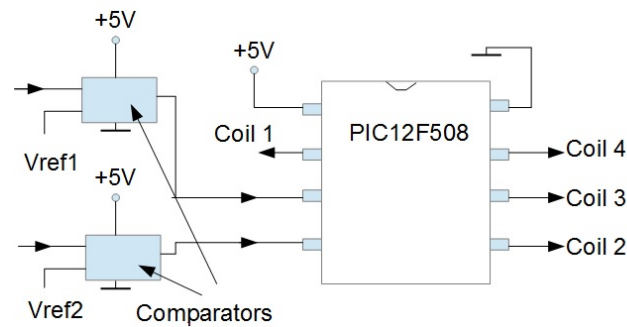


Figure 9. The schematic diagram of the PIC Microcontroller wiring. The electronic comparators are used to tune our digital controller; so, the voltages V_{ref1} and V_{ref2} are manually adjusted by mechanical potentiometers.

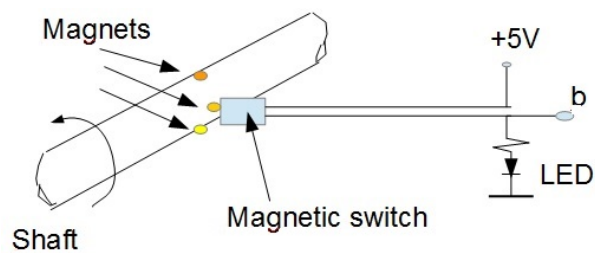


Figure 10. The electronic realization for the shaft velocity measurement. There are 4 permanent magnets located along the shaft diameter, and the magnetic switch is the one from the Cherry firm: MP201901(1316B). Point *b* is the sensor output.

3. Digital control design for vibration reduction on a journal magnetic bearing

The digital control is easy to visualize from the student control engineering point of view: Each coil is activate if the corresponding proximity sensor is not set. This is realized and shown in the program in Fig. 11. This program is written by using the assembler code: the low level program of microcontroller devices. The obtained experiment result is evidenced in Fig. 12. Finally, and from the shaft velocity sensor, the corresponding velocity, with or without the controller, was approximately 2143 rpm.

```
list    p=12F508 ; list directive
define processor
#include "p12f508.inc" ;
processor specific variable
definitions
;_CONFIG _OSC_IntRC &
_WDT_OFF & _CP_OFF &
_MCLR_OFF
;(4MHz Internal Clock)
AUX1 EQU 07H
ORG 0000H
; Port configuration
    MOVLW b'00011000'
    TRIS 6
; Timer configuration
    MOVLW b'11010111'
    OPTION
INI NOP
;Magnetic actuator 1
    BTFSS GPIO,04H
    BSF GPIO,05H
    BTFSC GPIO,04H
    BCF GPIO,05H
;....
;Magnetic actuator 3
    BTFSS GPIO,04H
    BCF GPIO,01H
    BTFSC GPIO,04H
    BSF GPIO,01H
;....
;Magnetic actuator 2
    BTFSS GPIO,03H
    BSF GPIO,02H
    BTFSC GPIO,03H
    BCF GPIO,02H
;....
;Magnetic actuator 4
    BTFSS GPIO,03H
    BCF GPIO,00H
    BTFSC GPIO,03H
    BSF GPIO,00H
;.... ZEROS
    BCF GPIO,00H
    BCF GPIO,02H
    BCF GPIO,01H
    BCF GPIO,05H
    GOTO INI
END
```

Figure 11. The assembler program of the proposed digital control to vibration mitigation of the corresponding journal bearing. The 'zeros' programming part serves as a kind of PWM signal control generator to the coil drivers; an important issue to keep the coils temperatures to safety levels.

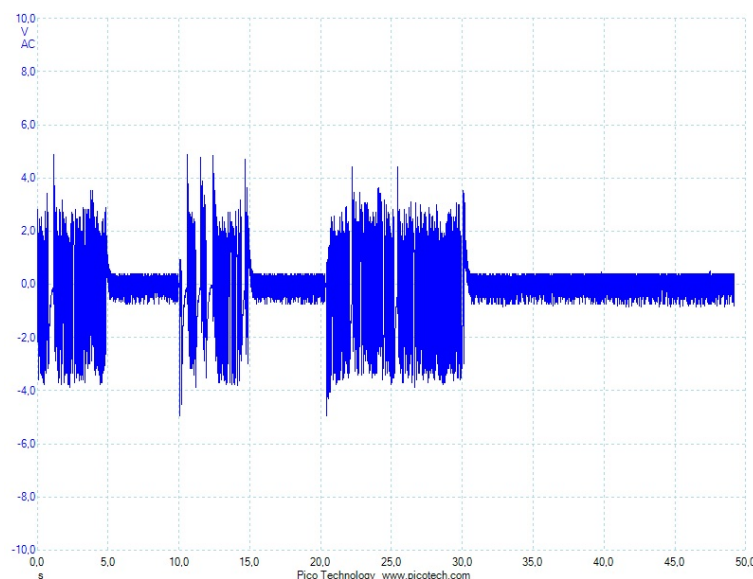


Figure 12. Experimental result. Vibration on the shaft vertical movement (measurement at the corresponding point *a* in Fig. 8). The controller is approximately activated at 5s, 15s, and 30s; and approximately deactivated at 0, 10s, and 20s. This signal was obtained using the PicoScope data acquisition card in AC mode operation.

4. Conclusions

In this paper, an easy digital control to magnetically manipulated a journal bearing was designed. The controller was programmed by using a cheap microcontroller, and by employing the assembler programming language. According to our experiments, this controller was able to mitigate vibration on a imbalance rotating shaft. We believe that this platform may be illustrative to teach control and mechatronics to engineering students. Obviously, on this experiment, it is assumed that the student has previous acknowledge on basic electronics design, mechanical concepts, and on microcontroller theory including the assembler code, and so on.

Acknowledgments: This work was partially supported by the Spanish Ministry of Economy and Competitiveness under grant number DPI2015-64170-R (MINECO/FEDER).

Author Contributions: The authors contributed equally to this work.

Conflicts of Interest: The authors declare no conflict of interest.

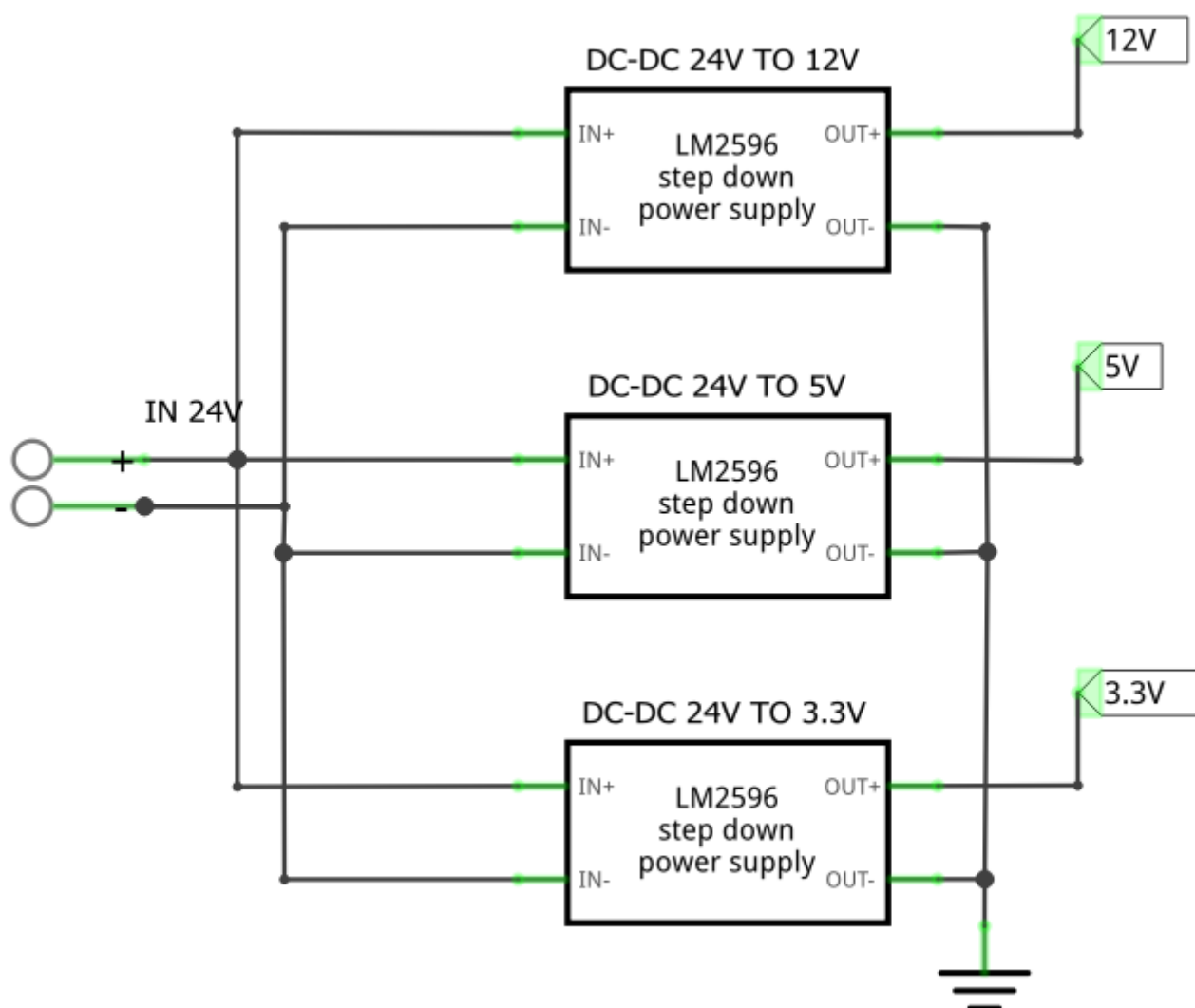
Bibliography



1. Zhou, S.; Shi, J. Active balancing and vibration control of rotating machinery: a survey. *Shock and Vibration Digest* **2001**, *33*, 361–371.
2. Kryszynski, T.; Malburet, F. *Mechanical vibrations: active and passive control*; Vol. 103, John Wiley & Sons, 2010.
3. Allaire, P.; Lewis, D.; Knight, J. Active vibration control of a single mass rotor on flexible supports. *Journal of the Franklin Institute* **1983**, *315*, 211–222.
4. Chen, W.J.; Gunter, E.J. *Dynamics of rotor-bearing systems*; Trafford publishing, Canada, 2010.
5. Luo, N.; Vidal, Y.; Acho, L. *Wind Turbine Control and Monitoring*; Springer, 2014.
6. Arias-Montiel, M.; Silva-Navarro, G.; Antonio-García, A. Active Vibration Control in a Rotor System by an Active Suspension with Linear Actuators. *Journal of applied research and technology* **2014**, *12*, 898–907.

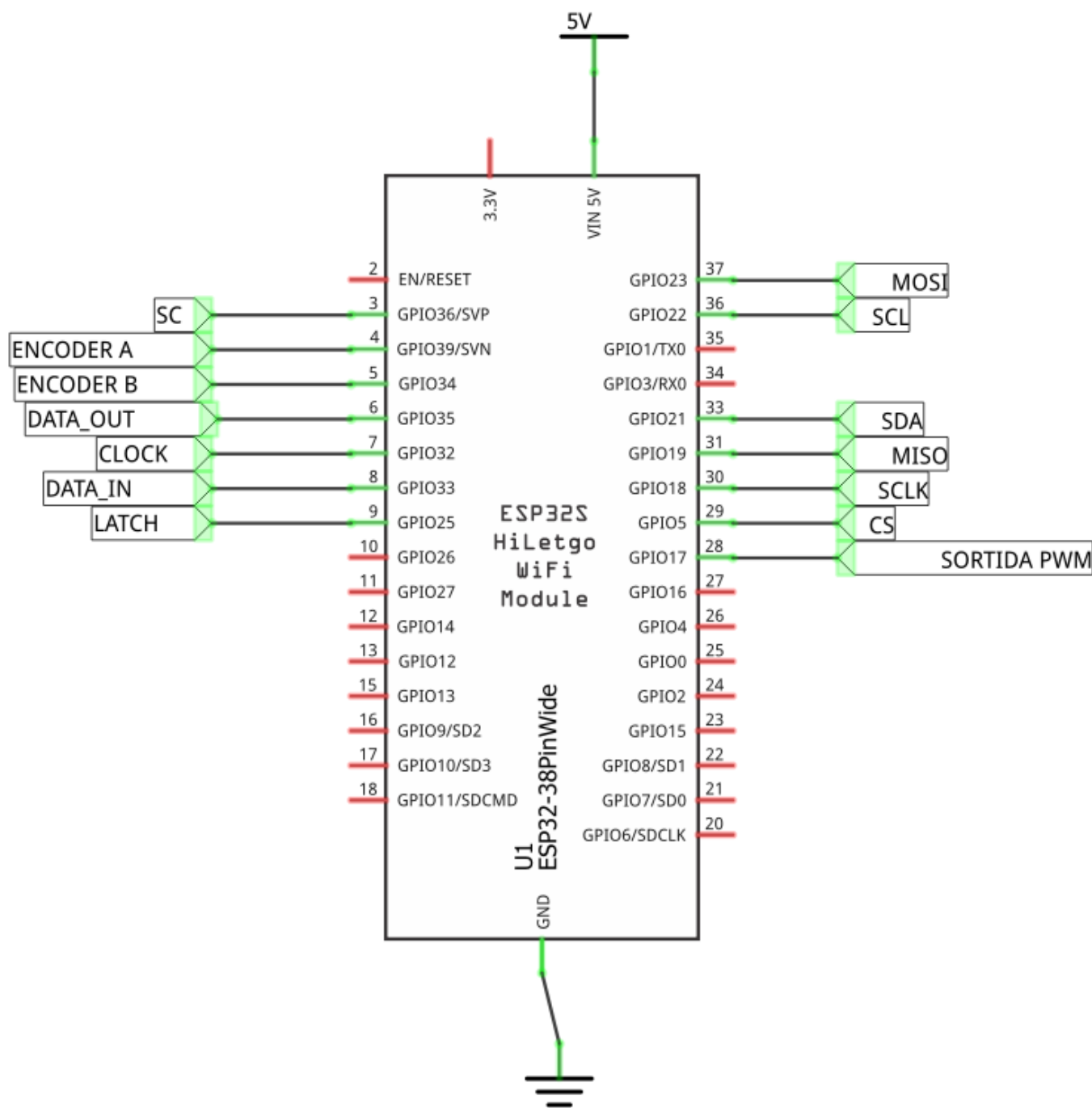
- 124 7. Chasalevris, A.; Dohnal, F. Improving stability and operation of turbine rotors using adjustable journal
125 bearings. *Tribology International* **2016**, *104*, 369–382.
- 126 8. Bab, S.; Khadem, S.; Shahgholi, M.; Abbasi, A. Vibration attenuation of a continuous rotor-blisk-journal
127 bearing system employing smooth nonlinear energy sinks. *Mechanical Systems and Signal Processing* **2017**,
128 *84*, 128–157.
- 129 9. Humphris, R.; Kelm, R.; Lewis, D.; Allaire, P. Effect of control algorithms on magnetic journal bearing
130 properties. *Journal of Engineering for Gas Turbines and Power* **1986**, *108*, 624–632.
- 131 10. El-SHAFEI, A.; DIMITRI, A. Controlling journal bearing instability using active magnetic bearings
132 [C/CD]. *Proceeding of A SME Turbo Expo* **2007**, pp. 14–17.
- 133 11. El-Shafei, A.; Dimitri, A. Controlling journal bearing instability using active magnetic bearings. *Journal*
134 *of Engineering for Gas Turbines and Power* **2010**, *132*, 012502.
- 135 12. Knospe, C.R.; Fedigan, S.J.; Hope, R.W.; Williams, R.D. A multitasking dsp implementation of adaptive
136 magnetic bearing control. *IEEE Transactions on control systems technology* **1997**, *5*, 230–238.
- 137 13. Predko, M. *Programming and Customizing the PC Microcontroller*; McGraw-Hill, Inc., 1997.
- 138 14. Gamazo-Real, J.C.; Vázquez-Sánchez, E.; Gómez-Gil, J. Position and speed control of brushless DC motors
139 using sensorless techniques and application trends. *Sensors* **2010**, *10*, 6901–6947.



140 © 2016 by the authors. Submitted to *Energies* for possible open access publication under the terms and
141 conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>)

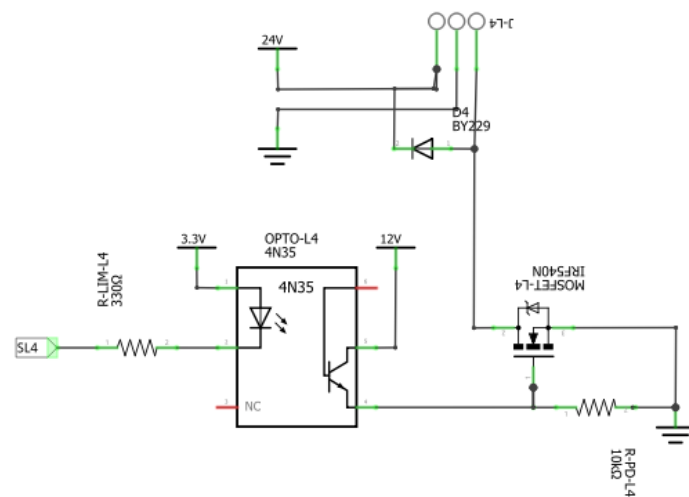
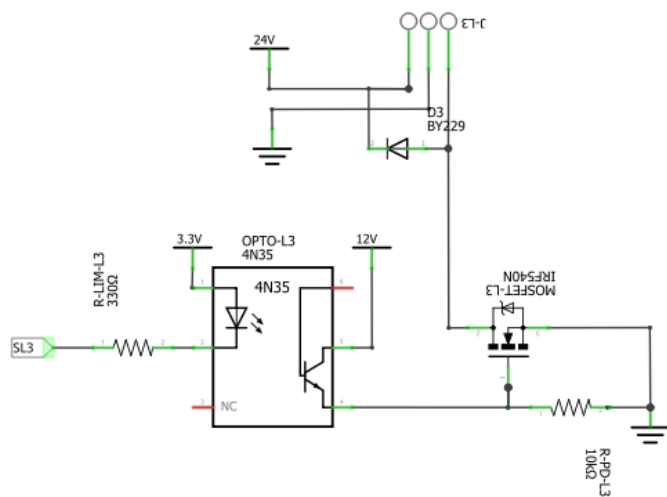
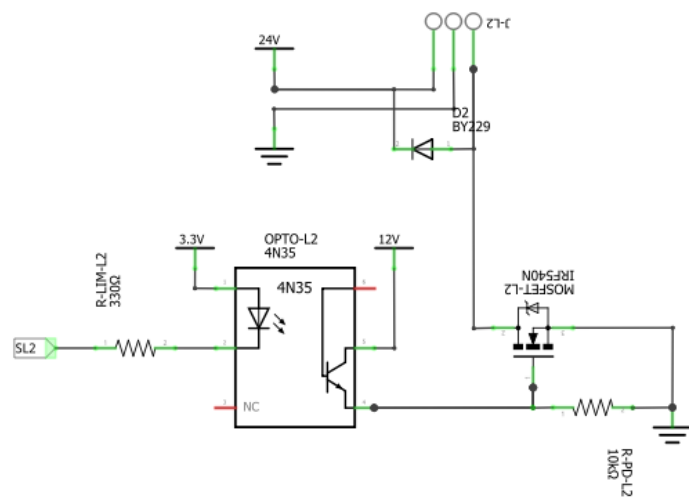
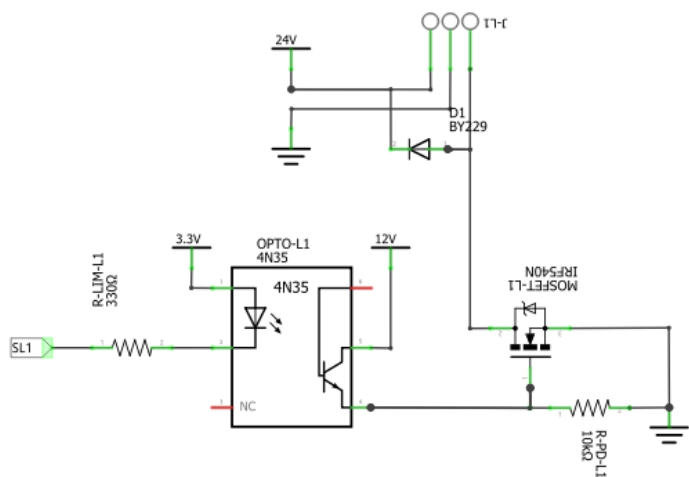
8.2. Annex 2: Esquemàtics





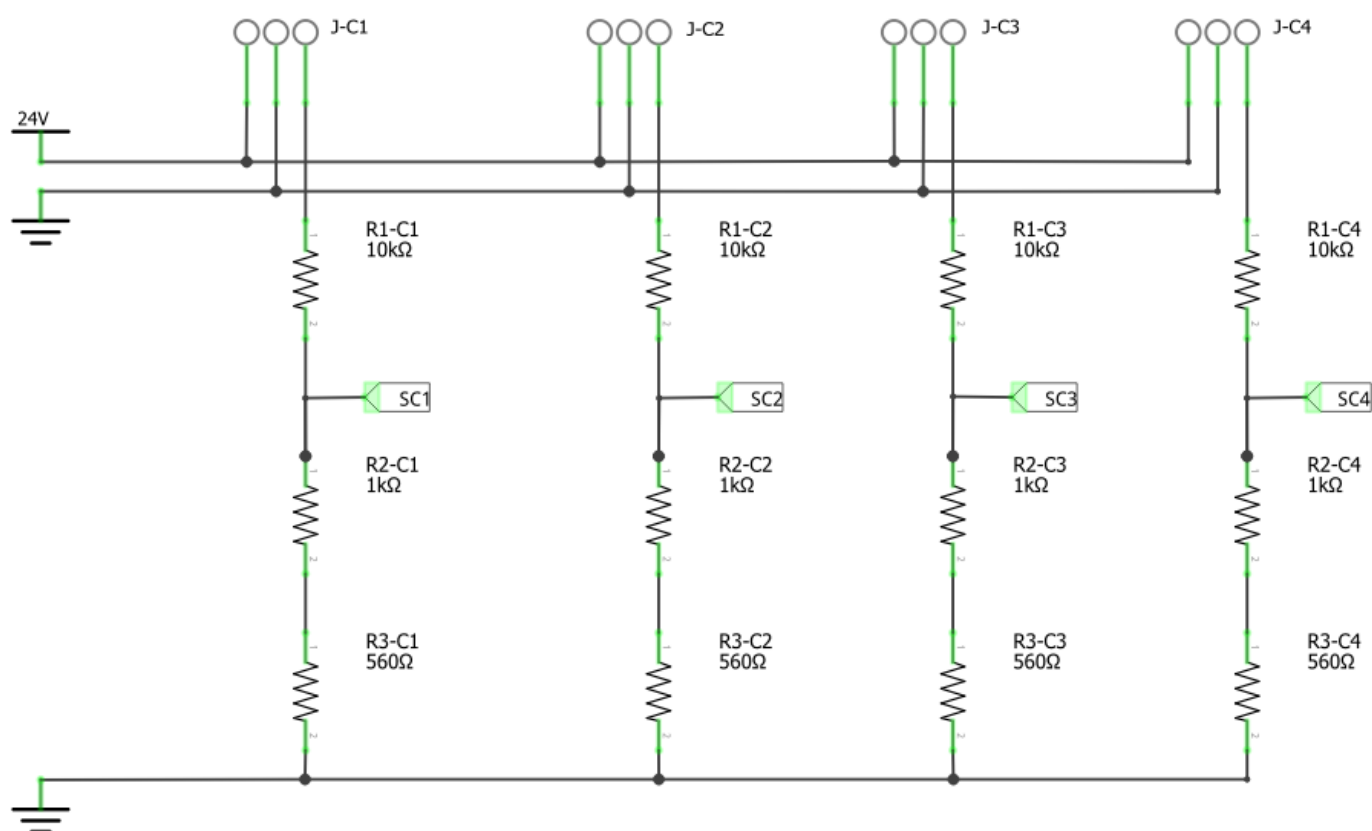
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	1
Document:	Esquemàtic Fonts d'alimentació
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 





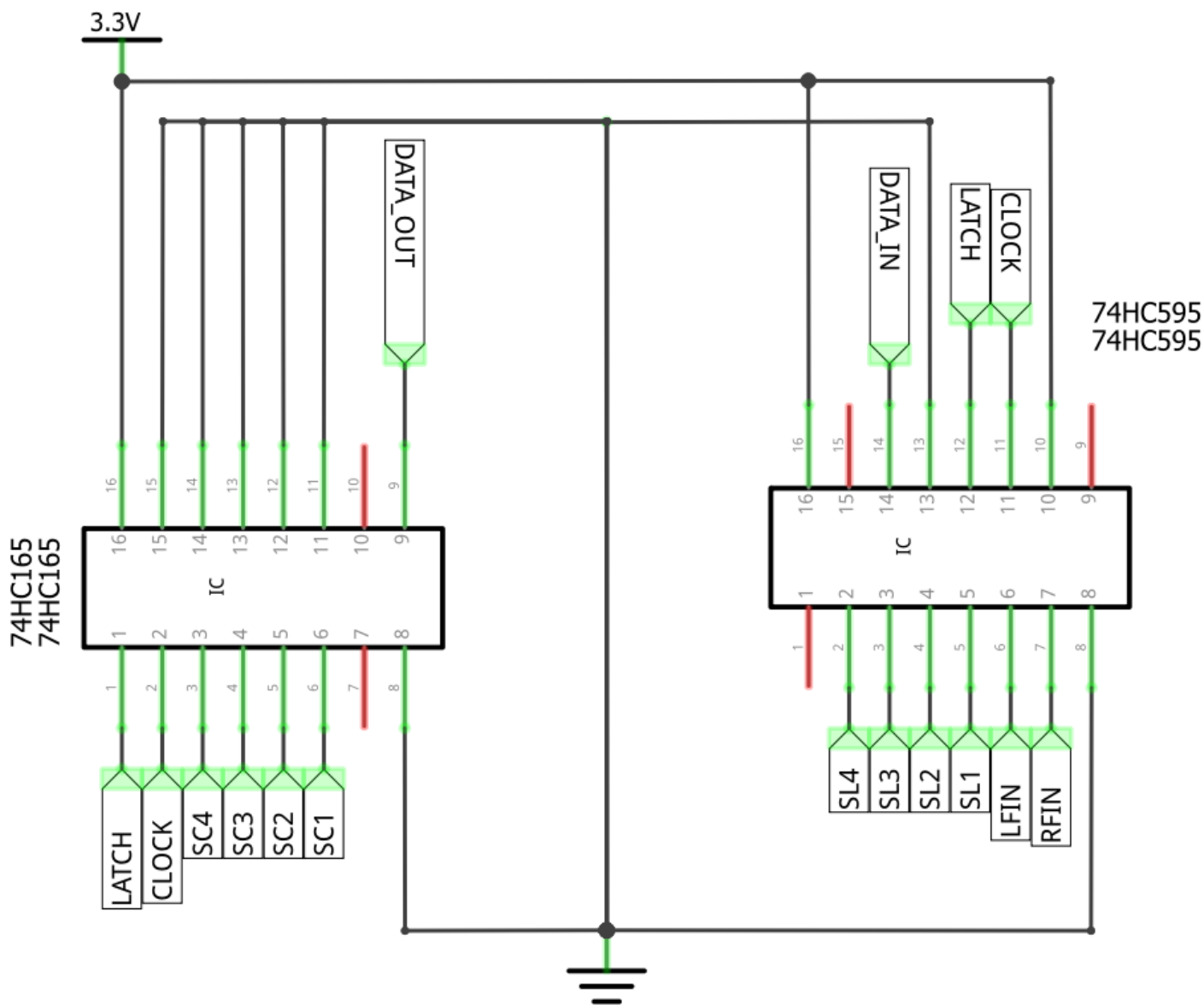
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	2
Document:	Esquemàtic Microcontrolador
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 





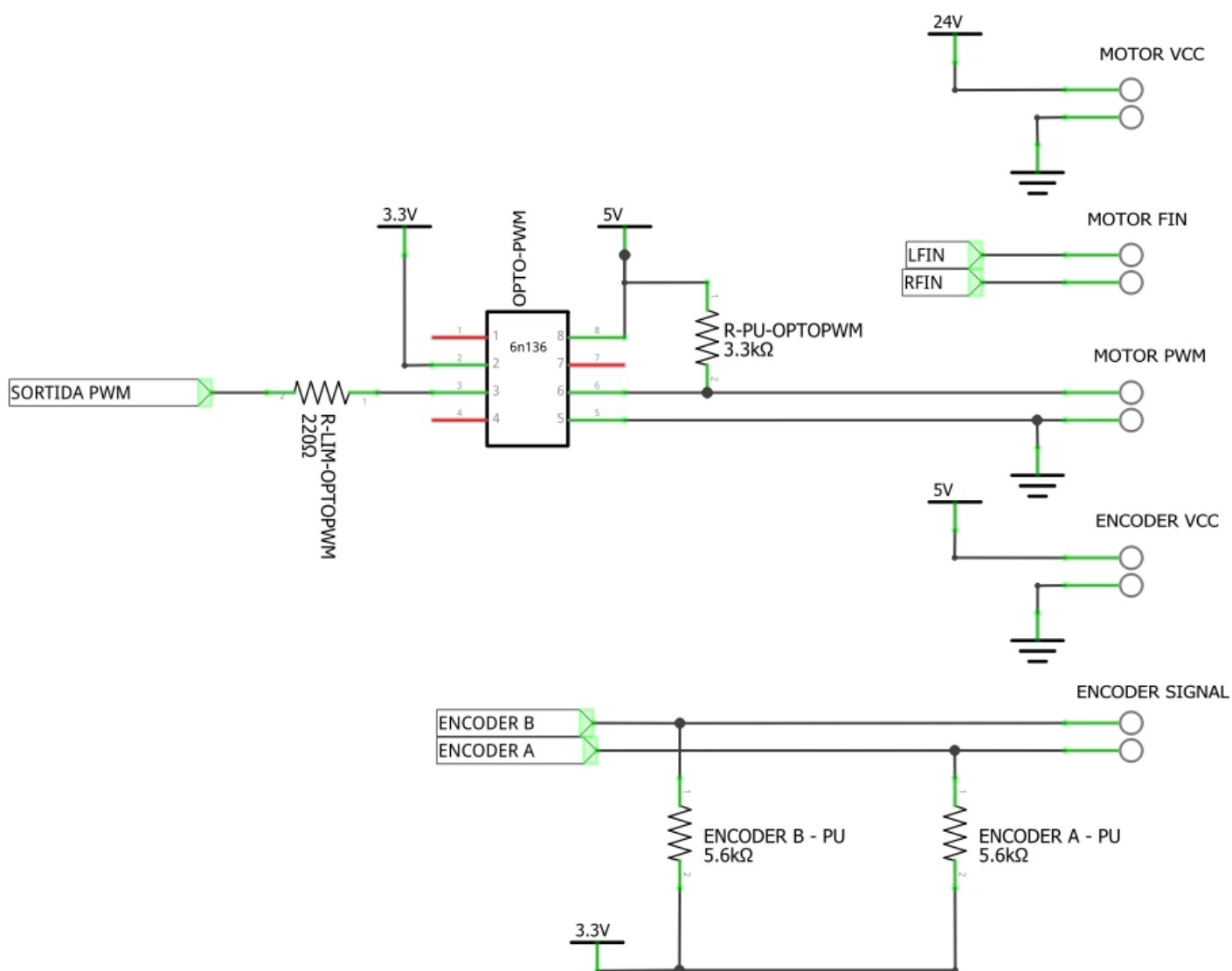
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	3
Document:	Esquemàtic Imants
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 





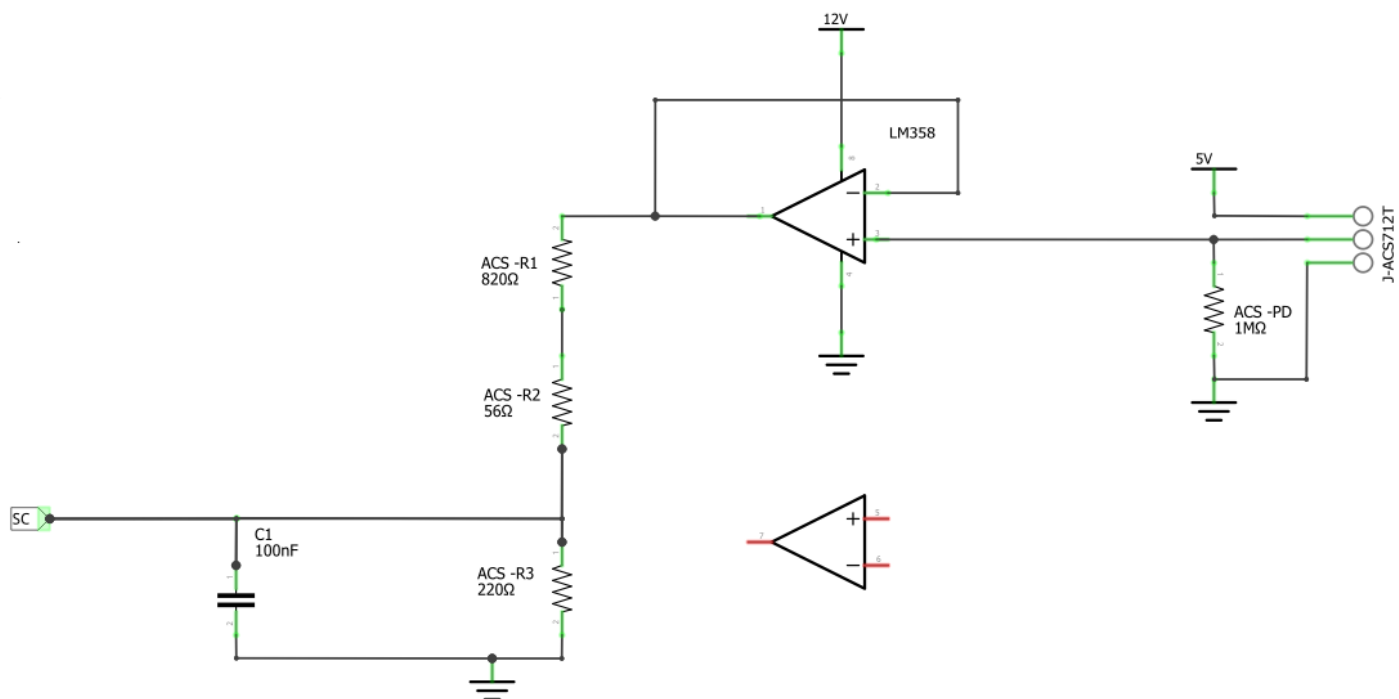
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	4
Document:	Esquemàtic Sensors
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 





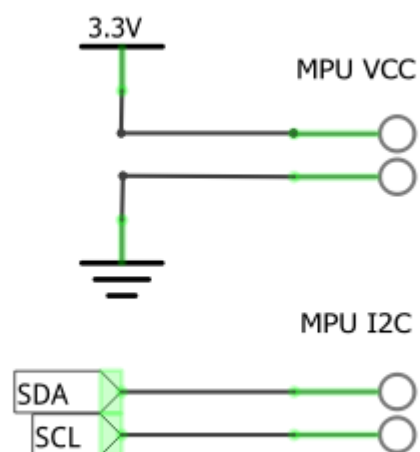
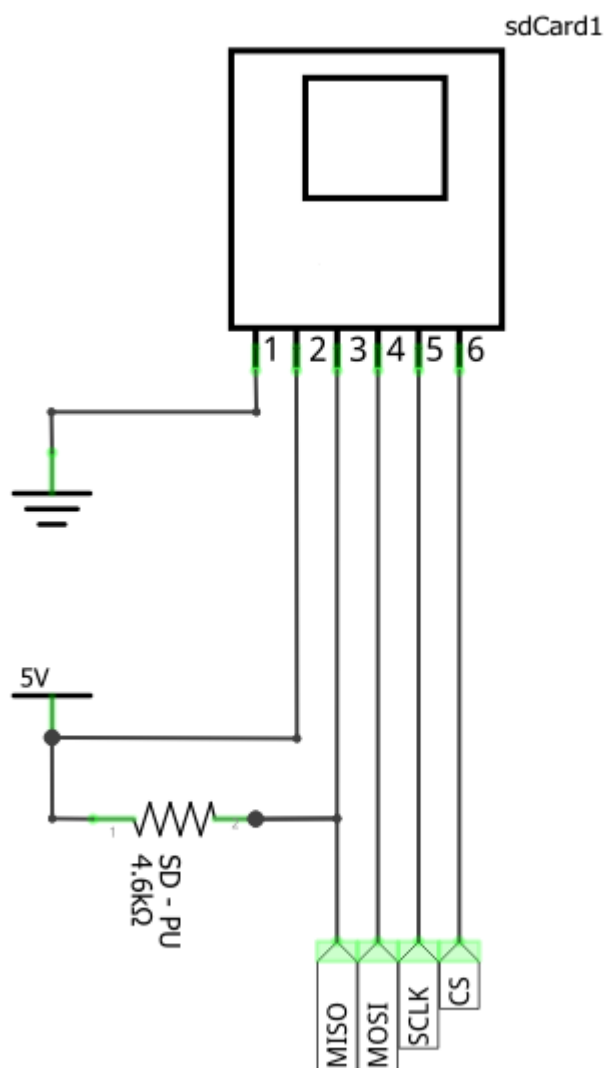
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	5
Document:	Esquemàtic 74HC595&74HC165
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 





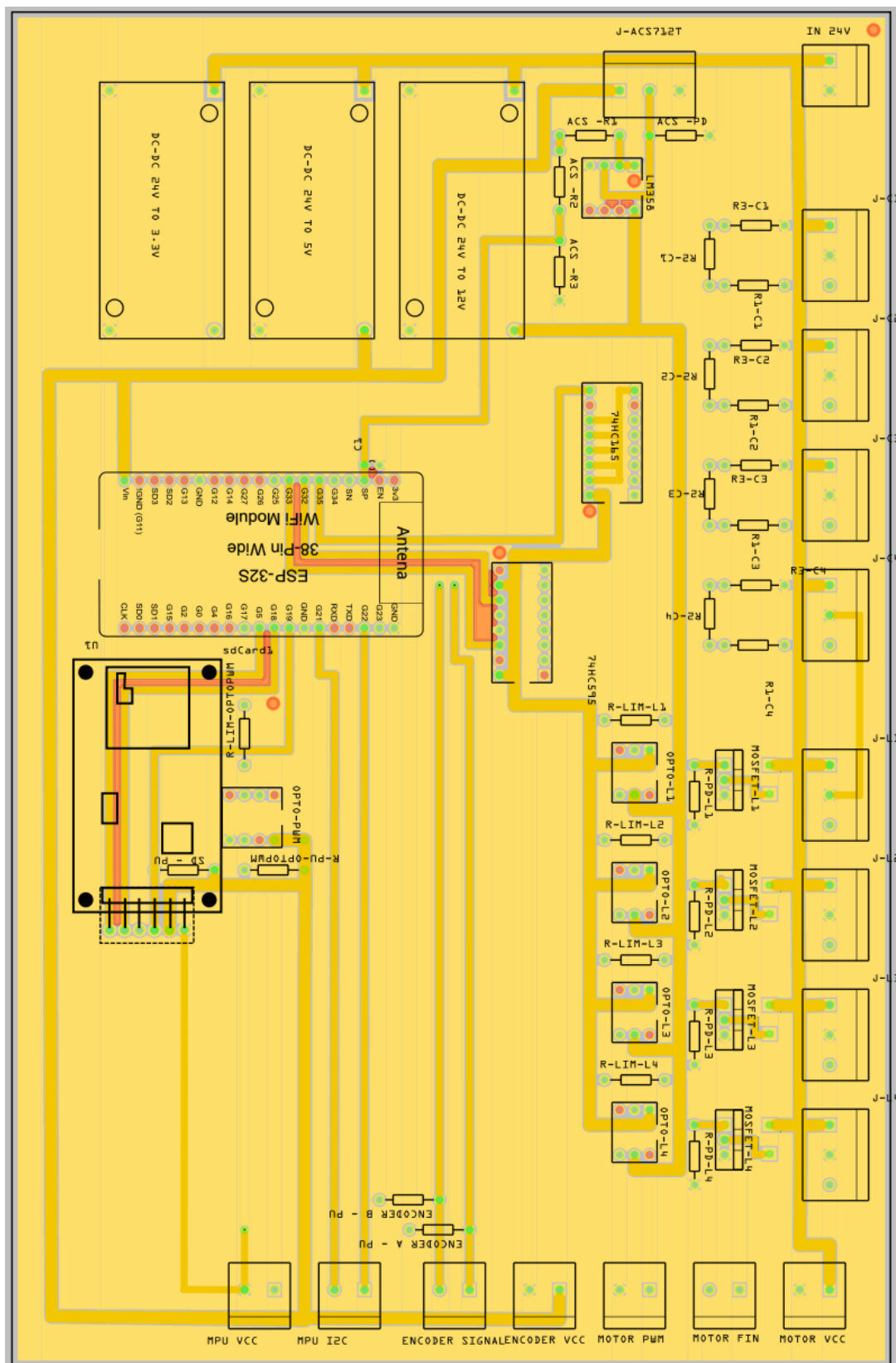
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	6
Document:	Esquemàtic Motor
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 




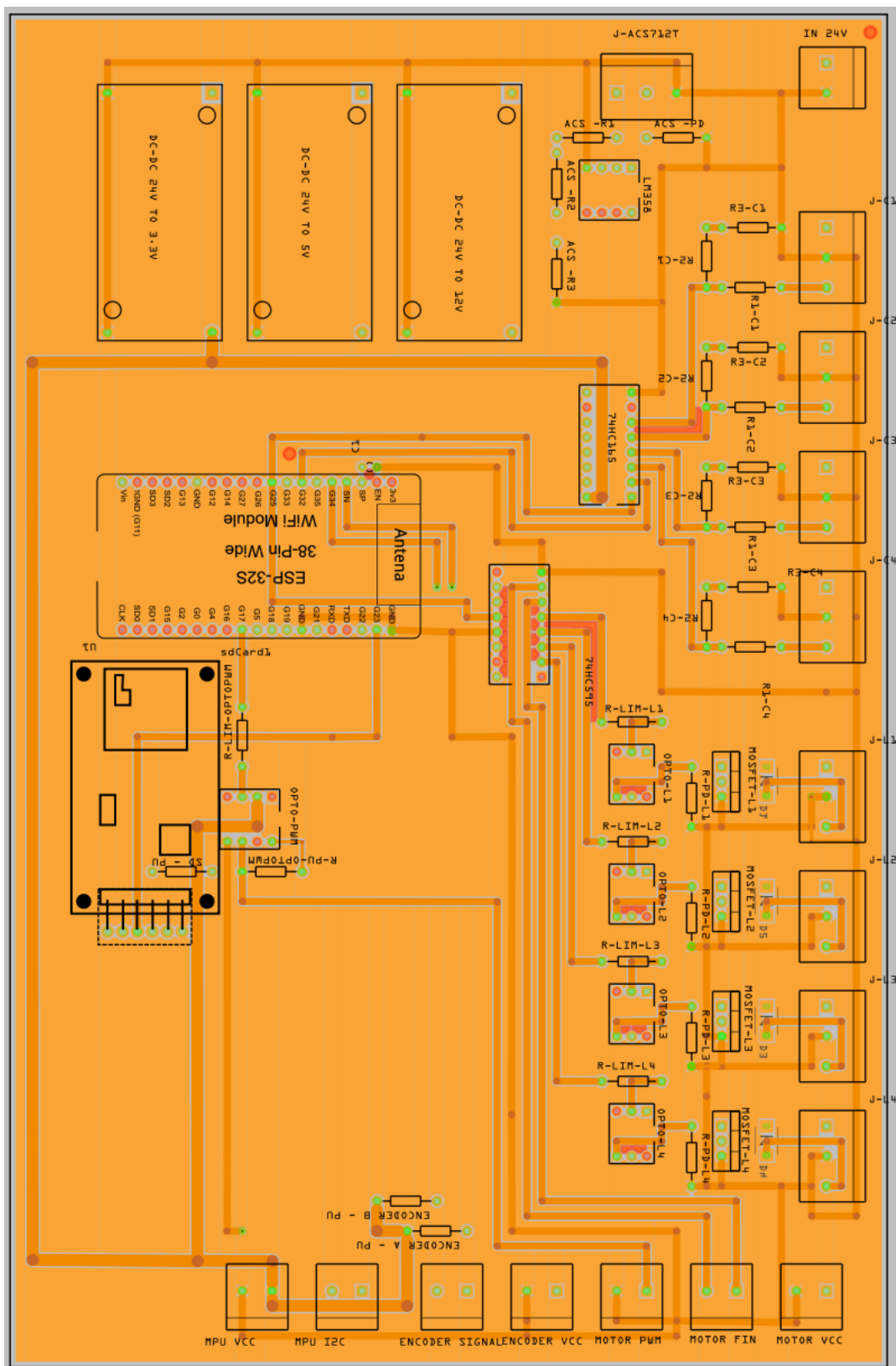
Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	7
Document:	Esquemàtic Sensor Corrent
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 






Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	8
Document:	Esquemàtic Lector Micro SD i Acceleròmetre
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa 



Data i Lloc:	05/07/2019, Manresa
Nombre de plànot:	9
Document:	PCB Superior
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH Escola Politècnica Superior d'Enginyeria de Manresa



Data i Lloc:	05/07/2019, Manresa
Nombre de plànol:	1
Document:	Esquemàtic Fonts d'alimentació
Autor:	Eduard Badia Claret
Treball Final de Grau: Control per a la levitació magnètica d'un eix	  

8.3. Annex 3: Codi de Software

```

/***** INCLUDES *****/
#include <Arduino.h>
/*WIFI*/
#include <WiFi.h>
#include <WiFiAP.h>
#include <ESP32WebServer.h>
#include <ESPmDNS.h>
#include <ArduinoJson.h>
/*TIMER*/
#include <Ticker.h>
/*ACCELEROMETRE*/
#include <Wire.h> //I2C 22=SCL 21=SDA
#include <MPU6050_tockn.h>
/*SENSOR CURRENT*/
#include <driver/adc.h>
#include "esp_adc_cal.h"
/*GESTIÓ FITXER*/
#include "SdFat.h"
#include "SPI.h"
/*MOTOR*/
#include <PID_v1.h>
/*WATCHDOG*/
#include "esp_system.h"
/***** DEFINES *****/
#define PLACA 0 //1 = PROTO, else = BONA
#define DEVELOP false
#if DEVELOP
#define DEV_ACC false
#define DEV_WIF false
#define DEV_MAG false
#define DEV_CUR false
#define DEV_FILE false
#define DEV_JSON false
#define DEV_SD false
#define DEV_MOT false
#define DEV_MOT2 false
#endif
/*WIFI*/
#define servername "maqueta"
/*TIMER*/
#define INTERRUPT 1 //ms
/*ACCELEROMETRE*/
/*74HC165 (IN) 74HC595 (OUT)*/
#define LATCH 25
#define CLOCK 32
#define DATA_IN 35
#define DATA_OUT 33
/*SENSOR CURRENT*/
#define SC 36
#define VREF 1128
#define MV_PER_AMP 100
#define ACS_OFFSET 520
#define N_READS 100
/*GESTIÓ FITXER*/
#define SD_CHIP_SELECT SS
#define MAX_ARRAY 30
/*MOTOR*/
#define encoderPinA 39
#define encoderPinB 34
#define sortidaPWM 17
#define COUNTSPERREVOLUTION 28.0
#define QUADRATURE 4.0
#define RPMMOTORMAX 1420.0
#define FREQ 5000
#define PWM 0
#define RESOLUTION 10

/***** GLOBAL VARS *****/
/*WIFI*/
ESP32WebServer server(80);
IPAddress IP(192, 168, 14, 14);
IPAddress gateway(192, 168, 14, 14);
IPAddress mask = (255, 255, 255, 0);
IPAddress dns(192,168,14,1);
const char* ssid = "maqueta";
const char* passwd = "abcd1234";
uint8_t max_connection = 1;
String webpage = "";
String header;
String motorState = "OFF", imantsState = "OFF";
uint8_t fileState = 0;
/*TIMER*/
Ticker ticker;
/*ACCELEROMETRE*/
MPU6050 mpu6050(Wire);
/*SENSOR CURRENT*/
double Amps = 0.0;
esp_adc_cal_characteristics_t adc_chars;
/*GESTIÓ FITXER*/
#define ServerVersion "1.0"
SdFat sd;
SdFile mainRoot, newRoot;
File fileW;
uint8_t stateFile = 0x00;
String filename = "";
uint32_t millisFitxer = 0;
int16_t rpm = 0;
uint8_t num = 0;
uint16_t numFolder = 0, numFile = 0;
/*74HC165 (IN) 74HC595 (OUT)*/
uint8_t readByte = 0x00, writeByte = 0x7E;
/*MOTOR*/
int16_t oldEncoderValue = 0, encoderValue = 0, cm = 0, encoderValueOld = 0;
uint16_t tempsPols = 0;

```



```

double Input = 0.0, Output = 0.0, workSetpoint = 0.0, sensorSetpoint = 0.0;
float Ki = 1, Kp = 1.4, Kd = 0.00;
bool motorOn = false, imantsOn = false, newValues = true;
PID myPID(&Input, &Output, &workSetpoint, Kp, Ki, Kd, P_ON_E, DIRECT);

/***** CODE *****/
/*----- SETUP -----*/

void setup() {
  #if DEVELOP
    // Serial Port
    Serial.begin(115200);
    Serial.println("Serial Port Working OK");
  #endif
  /*GESTIÓ FITXER*/
  initSD();
  /*WIFI*/
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password, 1, 0, max_connection);
  WiFi.softAPConfig(IP, IP, mask);
  if (!MDNS.begin(servername)) {          // Set your preferred server name, if you use "myserver" the address would be http://myserver.local/
    ESP.restart();
  }
  server.on("/", initialWebSite);
  server.on("/download", SendFile);
  server.on("/dir", Directory);
  server.on("/downordel", File_DownOrDel);
  server.on("/FILE/started", FileStart);
  server.on("/FILE/stopped", FileStop);
  server.on("/motor/on", MotorOn);
  server.on("/motor/off", MotorOff);
  server.on("/imants/on", ImantsOn);
  server.on("/imants/off", ImantsOff);
  server.on("/setParams", HTTP_POST, setParams);

  server.begin();

  #if DEV_WIF
    Serial.println(".");
    delay(500);
    Serial.println();
    Serial.println("Server started.");
    Serial.print("IP: ");    Serial.println(WiFi.softAPIP());
    Serial.print("MAC:");    Serial.println(WiFi.softAPmacAddress());
  #endif
  /*TIMER*/
  ticker.attach_ms(INTERRUPT, TimerInterrupt);
  /*ACCELEROMETRE*/
  Wire.begin();
  mpu6050.begin();
  #if DEV_ACC
    mpu6050.calcGyroOffsets(true);
  #else
    mpu6050.calcGyroOffsets();
  #endif
  /*74HC165 (IN) 74HC595 (OUT)*/
  pinMode(LATCH, OUTPUT);
  pinMode(CLOCK, OUTPUT);
  pinMode(DATA_IN, INPUT);
  pinMode(DATA_OUT, OUTPUT);
  digitalWrite(CLOCK, LOW);
  digitalWrite(LATCH, HIGH);
  /*MOTOR*/
  pinMode(encoderPinA, INPUT);
  digitalWrite(encoderPinA, HIGH);
  pinMode(encoderPinB, INPUT);
  digitalWrite(encoderPinB, HIGH);
  // PWM Configuration
  pinMode(sortidaPWM, OUTPUT);
  ledcSetup(PWM, FREQ, RESOLUTION);
  ledcAttachPin(sortidaPWM, PWM);
  ledcWrite(PWM, 0);
  // PID Configuration
  myPID.SetMode(MANUAL);
  myPID.SetOutputLimits(-1023, 1023);
  // Encoder Interrupts
  attachInterrupt(digitalPinToInterrupt(encoderPinA), doEncoderA, CHANGE);
  attachInterrupt(digitalPinToInterrupt(encoderPinB), doEncoderB, CHANGE);
  /*SENSOR CURRENT*/
  adc1_config_width(ADC_WIDTH_BIT_9);
  adc1_config_channel_atten(ADC1_CHANNEL_0, ADC_ATTEN_DB_0);
  esp_adc_cal_characterize(ADC_UNIT_1, ADC_ATTEN_DB_0, ADC_WIDTH_BIT_9, VREF, &adc_chars);
}

/*----- LOOP -----*/

void loop() {
  server.handleClient();
  UpdateMotor();
  mpu6050.update();
  ReadCurrent();
  UpdateFile();
  #if DEV_ACC
    mpu6050Test();
  #endif
}

void UpdateMotor(void){
  static bool sentit = true;
  static int count = 0;
  if (motorOn == true){
    myPID.Compute();
    if (Output < 0){
  #if PLACA == 1 //PROTO

```

```

        writeByte &= 0xBD; //Amb placa proto
    #else //BONA
        writeByte &= 0x3F; //Amb placa bona
    #endif
        writeByte |= 0x40;
        ledcWrite(PWM, abs(Output));
    }else if (Output > 0){
    #if PLACA == 1 //PROTO
        writeByte &= 0xBD; // Amb placa proto
        writeByte |= 0x02; // Amb placa proto
    #else //BONA
        writeByte &= 0x3F; // Amb placa bona
        writeByte |= 0x80; // Amb placa bona
    #endif
        ledcWrite(PWM, abs(Output));
    } else{
    #if PLACA == 1 //PROTO
        writeByte &= 0xBD; // Amb placa proto
        writeByte |= 0x42; // Amb placa proto
    #else //BONA
        writeByte &= 0x3F; // Amb placa bona
        writeByte |= 0xC0; // Amb placa bona
    #endif
        ledcWrite(PWM, 0);
    }
    } else {
    #if PLACA == 1 //PROTO
        writeByte &= 0xBD; // Amb placa proto
        writeByte |= 0x42; // Amb placa proto
    #else //BONA
        writeByte &= 0x3F; // Amb placa bona
        writeByte |= 0xC0; // Amb placa bona
    #endif
        ledcWrite(PWM, 0);
    }
    }
    #if DEV_MOT
    if ((count % 100) == 0){
        if (motorOn == true){
            Serial.print("Setpoint: ");
            Serial.print(workSetpoint);
            Serial.print(" - INPUT: ");
            Serial.println(Input);
            Serial.print(" - OUTPUT: ");
            Serial.println(Output);
        }
    }
    count++;
    #endif
}

// Encoder A Interrupt Function
void doEncoderA()
{
    if (digitalRead(encoderPinA) == digitalRead(encoderPinB)) {
        encoderValue--;
    } else {
        encoderValue++;
    }
}

// Encoder B Interrupt Function
void doEncoderB()
{
    if (digitalRead(encoderPinA) == digitalRead(encoderPinB)) {
        encoderValue++;
    } else {
        encoderValue--;
    }
}

void motorCalc (void){
    int16_t rpmMOTOR = 0;
    static uint8_t count = 0;

    count++;
    if ((count % 100) == 0){
        if (motorOn){
            rampCalculator();
            rpmMOTOR = ((encoderValue * 60.0 * 10) / COUNTSPERREVOLUTION) / QUADRATURE;
            Input = (rpmMOTOR * 1023.0) / RPMMOTORMAX;
            encoderValueOld = encoderValue;
            encoderValue = 0;
        } else {
            rpmMOTOR = 0;
            encoderValueOld = 0;
            encoderValue = 0;
        }
        count = 0;
    }

    #if DEV_MOT2
    Serial.print ("EncoderValue: ");
    Serial.print (encoderValueOld);
    Serial.print (" - RpmMotor:");
    Serial.print (rpmMOTOR);
    Serial.print (" - Input: ");
    Serial.print (Input);
    Serial.print (" - Output: ");
    Serial.print (Output);
    Serial.print (" - Setpoint: ");
    Serial.println (workSetpoint);
    #endif
}
}

```

```

void rampCalculator(void){
    if((workSetpoint+10 >= sensorSetpoint)&&(sensorSetpoint >= workSetpoint-10)){
        workSetpoint = sensorSetpoint;
    }else if(sensorSetpoint >= workSetpoint+200){
        workSetpoint += 50;
    }else if(sensorSetpoint <= workSetpoint-200){
        workSetpoint -= 50;
    }else if(sensorSetpoint >= workSetpoint+100){
        workSetpoint += 10;
    }else if(sensorSetpoint <= workSetpoint-100){
        workSetpoint -= 10;
    }else if(sensorSetpoint >= workSetpoint){
        workSetpoint += 5;
    }else if(sensorSetpoint <= workSetpoint){
        workSetpoint -= 5;
    }
}

void resetModule() {
    esp_restart();
}

void SendHTML_Content(void){
    server.sendContent(webpage);
    webpage = "";
}

void SendHTML_Stop(){
    server.sendContent("");
    server.client().stop(); // Stop is needed because no content length was sent
}

void SelectInput(String heading1){
    webpage += F("<h3>"); webpage += heading1 + "</h3>";
    webpage += F("Exemples:<br> - Per esborrar/descarregar fitxer dins d'una carpeta    '/1/1.txt'");
    webpage += F("<br> - Per esborrar una carpeta (cal que estigui buida)    '/1' <br><br>");
    webpage += F("<FORM action='\"/downordel\"' method='post'>"); // Must match the calling argument e.g. '/chart' calls '/chart' after selection but with
    webpage += F("<input type='text' name='\"'; webpage += \"Fitxer: \"; webpage += F(\"' value='\"><br>");
    webpage += F("<input type='submit' name='\"; webpage += \"download\"; webpage += F(\"' value='Descarregar'>");
    webpage += F("<input type='submit' name='\"; webpage += \"delete\"; webpage += F(\"' value='Esborrar'><br><br></form>");
}

void File_DownOrDel(){ // This gets called twice, the first pass selects the input, the second pass then processes the command line arguments
    if (server.args() > 0 ) { // Arguments were received
        if (server.hasArg("download")) DownloadFile(server.arg(0));
        if (server.hasArg("delete")) DeleteFile(server.arg(0));
    }
}

void Directory(void){
    SdFile root;
    root.open("/maqueta/");
    if (root.isDir()) {
        root.rewind();
        SendHTML_Header();
        webpage += F("<h1>Contingut carpeta \"Maqueta\"</h1><br>");
        webpage += F("<table class='\"table1\">");
        webpage += F("<tr><th>Nom</th><th style='width:20%'>Tipus</th><th>Mida</th></tr><br>");
        printDirectory("/maqueta/",0);
        webpage += F("</table>");
        SendHTML_Content();
        SelectInput("Escriu el nom del fitxer (amb l'extensió)");

        root.close();
    }
    else
    {
        SendHTML_Header();
        webpage += F("<h3>No Files Found</h3>");
    }
    append_page_footer();
    SendHTML_Content();
    SendHTML_Stop(); // Stop is needed because no content length was sent
}

void DownloadFile(String filename2){
    File localFile = sd.open("/maqueta/"+filename2);//, "r");
    #if DEV_SD
        Serial.println ("DOWN FILE");
        Serial.println (filename2);
    #endif

    if (localFile){
        if(!localFile.isDirectory()){
            #if DEV_SD
                Serial.println ("OBERT");
            #endif
            server.sendHeader("Content-Type", "text/text");
            server.sendHeader("Content-Disposition", "attachment; filename=" + filename2);
            server.sendHeader("Connection", "close");
            server.streamFile(localFile, "application/octet-stream");
            server.sendContent("");
            server.client().stop();
            localFile.close();
        } else ReportFileNotPresent("download");
    } else ReportFileNotPresent("download");
}

void DeleteFile(String inFile) { // Delete the file
    char buff[MAX_ARRAY];
    SdFile localFile;
    String in2File = "/maqueta/"+inFile;

    in2File.toCharArray(buff, (MAX_ARRAY));

```

```

    localFile.open((buff), O_RDONLY);
    SendHTML_Header();

    if(localFile.isDir()){
        localFile.close();
        if(sd.rmdir(buff)){
            Serial.println(("Folder deleted successfully"));
        }
    }
    webpage += F("<h1>Contingut carpeta \"Maqueta\"</h1><br>");
    webpage += F("<h3>La carpeta '"+inFile+"' ha estat esborrat correctament</h3><br>");
    }else {
        webpage += F("<h1>Contingut carpeta \"Maqueta\"</h1><br>");
        webpage += F("<h3>ERROR: La carpeta '"+inFile+"' NO ha estat esborrat</h3><br>");
    }
    }else{
        localFile.close();
        if(sd.remove(buff)){
            Serial.println(("File deleted successfully"));
        }
    }
    webpage += F("<h1>Contingut carpeta \"Maqueta\"</h1><br>");
    webpage += F("<h3>El fitxer '"+inFile+"' ha estat esborrat correctament</h3><br>");
    }else {
        webpage += F("<h1>Contingut carpeta \"Maqueta\"</h1><br>");
        webpage += F("<h3>ERROR: El fitxer '"+inFile+"' NO ha estat esborrat</h3><br>");
    }
    }

    append_page_footer();
    SendHTML_Content();
    SendHTML_Stop();
}

void ReportFileNotPresent(String target){
    Serial.println ("Report no present");
}

void printDirectory(const char * dirname, uint8_t levels){
    char FileName[16];
    uint8_t nTabs = 0, i = 0;
    SdFile root, file;

    nTabs = (0-levels);
    Serial.print ("Directori: ");
    Serial.print (dirname);
    Serial.print (" - Nivell1: ");
    Serial.println ((nTabs));
    root.open(dirname);
    while(file.openNext(&root, O_RDONLY)){
        file.getName(FileName, 16);
        Serial.print("FileName : ");
        file.printName(&Serial);
        Serial.println();
        if (webpage.length() > 1000) {
            SendHTML_Content();
        }
        if(file.isDir()){
            webpage += "<tr><td>";
            if(nTabs > 0){
                for (i=0; i<nTabs; i++){
                    webpage += "--> ";
                }
            }
            webpage += String(FileName)+"</td><td>"+String(file.isDir())?"Dir":"File"+"</td><td></td></tr>";
            String folderfileName = String(dirname)+String(FileName);
            folderfileName.toCharArray(FileName, 16);
            printDirectory((FileName), levels-1);
        }
        else
        {
            webpage += "<tr><td>";
            if(nTabs > 0){
                for (i=0; i<nTabs; i++){
                    webpage += "--> ";
                }
            }
            webpage += String(FileName)+"</td>";
            webpage += "<td>"+String(file.isDir())?"Dir":"File"+"</td>";
            int bytes = file.fileSize();
            String fsize = "";
            if (bytes < 1024) fsize = String(bytes)+" B";
            else if(bytes < (1024 * 1024)) fsize = String(bytes/1024.0,3)+" KB";
            else if(bytes < (1024 * 1024 * 1024)) fsize = String(bytes/1024.0/1024.0,3)+" MB";
            else fsize = String(bytes/1024.0/1024.0/1024.0,3)+" GB";
            webpage += "<td>"+fsize+"</td></tr>";
        }
        file.close();
    }
}

/*----- WIFI -----*/

```

```

void append_page_header() {
webpage += F("<!DOCTYPE html><html>");
webpage += F("    <head>");
webpage += F("        <meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\"/>");
webpage += F("        <title>Control Maqueta</title>");
webpage += F("        <meta name=\"viewport\" content=\"width=device-width, initial-scale=1\"/>");
webpage += F("        <link rel=\"icon\" href=\"data:,\">
```

```

if (imantsState == "OFF") {
    webpage += F("
        <td><p><a href=\"\"/>imants/on\"><buttons class=\"\"/>
    } else {
        webpage += F("
        <td><p><a href=\"\"/>imants/off\"><buttons class=\"\"/>
    }

switch(fileState){
case 0: //Waiting to Start
    webpage += F("
        <td><p><a href=\"\"/>
    webpage += F("
        </tr>\">
    webpage += F("
        </table>\">
    break;
case 1:
    webpage += F("
        <td><p><a href=\"\"/>
    webpage += F("
        </tr>\">
    webpage += F("
        </table>\">
    break;
case 2:
    webpage += F("
        <td><p><a href=\"\"/>
    webpage += F("
        </tr>\">
    webpage += F("
        </table>\">
    webpage += F("
        <p><a align =\"center\" href=\"\"/>
    break;
}

webpage += F("
    <br><br>\">

webpage += F("
    <form id=\"setParams\" action=\"\"/>
webpage += F("
    <table class=\"table1\" align=\"center\">
webpage += F("
    <tr>
webpage += F("
    <th>Nom variable</th>
webpage += F("
    <th>Valor Actual</th>
webpage += F("
    <th>Unitat</th>
webpage += F("
    <th>Valor desitjat</th>
webpage += F("
    <th>Min % Max</th>
webpage += F("
    </tr>
webpage += F("
    <tr>
webpage += F("
    <td>Temps dels polsos</td>
webpage += F("
    <td>
webpage += F("
    </td>
webpage += F("
    <td>ms</td>
webpage += F("
    <td><input type=\"number\" name=\"tempsPols\" min=\"0\" max=\"1000\" numberType=\"int\" placeholder=\"0\"/></td>
webpage += F("
    <td>0 % 1000</td>
webpage += F("
    </tr>
webpage += F("
    <tr>
webpage += F("
    <td>PID: SetPoint</td>
webpage += F("
    <td>
webpage += F("
    </td>
webpage += F("
    <td>Rpm</td>
webpage += F("
    <td><input type=\"number\" name=\"rpm\" min=\"-990\" max=\"990\" numberType=\"int\" placeholder=\"0\"/></td>
webpage += F("
    <td>-990 % 990</td>
webpage += F("
    </tr>
webpage += F("
    <tr>
webpage += F("
    <td>PID : Kp</td>
webpage += F("
    <td>
webpage += F("
    </td>
webpage += F("
    <td>-</td>
webpage += F("
    <td><input type=\"number\" step=\"0.01\" name=\"Kp\" min=\"0\" numberType=\"float\" placeholder=\"0\"/></td>
webpage += F("
    <td>>0</td>
webpage += F("
    </tr>
webpage += F("
    <tr>
webpage += F("
    <td>PID : Ki</td>
webpage += F("
    <td>
webpage += F("
    </td>
webpage += F("
    <td>-</td>
webpage += F("
    <td><input type=\"number\" step=\"0.01\" name=\"Ki\" min=\"0\" numberType=\"float\" placeholder=\"0\"/></td>
webpage += F("
    <td>>0</td>
webpage += F("
    </tr>
webpage += F("
    <tr>
webpage += F("
    <td>PID : Kd</td>
webpage += F("
    <td>
webpage += F("
    </td>
webpage += F("
    <td>-</td>
webpage += F("
    <td><input type=\"number\" step=\"0.01\" name=\"Kd\" min=\"0\" numberType=\"float\" placeholder=\"0\"/></td>
webpage += F("
    <td>>0</td>
webpage += F("
    </tr>
webpage += F("
    </table>
webpage += F("
    <p><p><input type=\"submit\" value=\"Actualitzar Valors\"/></p></p>
webpage += F("
    </form>
webpage += F("
    <br><br>
webpage += F("
    </body>
webpage += F("
    <script>
webpage += F("
    document.getElementById(\"setParams\").addEventListener(\"submit\", function (event){
webpage += F("
        event.preventDefault();
webpage += F("
        event.stopPropagation();
webpage += F("
        var formData = {};
webpage += F("
        Object.keys(document.querySelectorAll('#setParams input[type=\"number\"]')).map(function(key){
webpage += F("
            var input = document.querySelector('#setParams input[type=\"number\"][key]\");
webpage += F("
            switch(input.getAttribute(\"numbertype\")) {
webpage += F("
                case \"int\":
webpage += F("
                    formData[input.name] = parseInt(input.value, 10);
webpage += F("
                    break;
webpage += F("
                case \"float\":
webpage += F("
                    formData[input.name] = parseFloat(input.value);
webpage += F("
                    break;
webpage += F("
                default:
webpage += F("
                    alert(\"Error, els atributs input type number necessiten l'atribut numberType (int o float)\");
webpage += F("
            }

```

```

webpage += F("                }");
webpage += F("            },"");
webpage += F("            Object.keys(document.querySelectorAll('#setParams input[type=\"text\"]')).map(function (key){");
webpage += F("                var input = document.querySelector('#setParams input')[key];");
webpage += F("                formData[input.name] = input.value");
webpage += F("            },"");
webpage += F("            fetch(\"http://192.168.14.14/setParams\", {");
webpage += F("                method: \"POST\",");
webpage += F("                body: JSON.stringify(formData)");
webpage += F("            });");
webpage += F("            .then(function(response){ if");
webpage += F("                (response.ok)");
webpage += F("                if(window.location.href.indexOf(\"refresh\") > -1){window.location = window.location.href;});");
webpage += F("                else {window.location = window.location.href + \"#refresh\";});");
webpage += F("                window.location.reload()");
webpage += F("            }");
webpage += F("        });");
webpage += F("    </script>");
webpage += F("    </html>");

append_page_footer();
}

void append_page_footer(){ // Saves repeating many lines of code for HTML page footers
webpage += F("<ul>");
if (fileState != 1){
    webpage += F("<li><a href='\">Home</a></li>");
    webpage += F("<li><a href='\">Directori</a></li>");
} else {
    webpage += F("<li><a>Home</a></li>");
    webpage += F("<li><a>Directori</a></li>");
}
webpage += F("</ul>");
webpage += "<footer>EPSEM UPC 2019 - TFG Eduard Badia Claret</footer>";
}

void initialWebSite (void){
    fileState = 0x00;
    UpdateClient();
}

void FileStart (void){
    fileState = 0x01; // File Starts, Writing untill Stop is pushed
    UpdateClient();
}

void FileStop (void){
    fileState = 0x02; // File Stops, is ready for being sent
    UpdateClient();
}

void MotorOn (void){
    motorOn = true;
    myPID.SetMode(AUTOMATIC);
    motorState = "ON"; // File Starts, Writing untill Stop is pushed
    UpdateClient();
}

void MotorOff (void){
    motorOn = false;
    Input = 0;
    Output = 0;
    workSetpoint = 0;
    myPID.SetMode(MANUAL);
    motorState = "OFF"; // File Starts, Writing untill Stop is pushed
    UpdateClient();
}

void ImantsOn (void){
    imantsOn = true;
    imantsState = "ON"; // File Starts, Writing untill Stop is pushed
    UpdateClient();
}

void ImantsOff (void){
    imantsOn = false;
    imantsState = "OFF"; // File Starts, Writing untill Stop is pushed
    UpdateClient();
}

void setParams(void){
    uint16_t tempTempsPols = NULL;
    int16_t tempRpm = NULL;
    float tempKp = NULL, tempKi = NULL, tempKd = NULL;
    char dataJson[200];
    bool negative = false;

    String strDataJson = server.arg("plain");
    if ((strDataJson.indexOf("-") != -1) {
        strDataJson.replace("-", "");
        negative = true;
    }
}

#if DEV_JSON
    Serial.println(strDataJson);
#endif
strDataJson.toCharArray(dataJson, 200);
StaticJsonDocument<200> jsonDoc;
DeserializationError errorJson = deserializeJson(jsonDoc, dataJson);
#if DEV_JSON
    if(errorJson){
        Serial.print("Json Error with code: ");
        Serial.println(errorJson.c_str());
        return;
    }
#endif
tempTempsPols = jsonDoc["tempsPols"];

```

```

tempRpm = jsonDoc["rpm"];
if (negative == true) tempRpm *= -1;
tempKp = jsonDoc["Kp"];
tempKi = jsonDoc["Ki"];
tempKd = jsonDoc["Kd"];

#if DEV_JSON
Serial.print(tempTempsPolis);
Serial.print(" ");
Serial.print(tempRpm);
Serial.print(" ");
Serial.print(tempKp);
Serial.print(" ");
Serial.print(tempKi);
Serial.print(" ");
Serial.println(tempKd);
#endif

if (!jsonDoc["tempsPolis"].isNull()) {
    tempsPolis = tempTempsPolis;
}

if (!jsonDoc["rpm"].isNull()){
    sensorSetpoint = (tempRpm);
    rpm = tempRpm;
}

if (!jsonDoc["Kp"].isNull())Kp = tempKp;
if (!jsonDoc["Ki"].isNull())Ki = tempKi;
if (!jsonDoc["Kd"].isNull())Kd = tempKd;

if ((tempKp != NULL) || (tempKi != NULL) || (tempKd != NULL)) myPID.SetTunings(Kp, Ki, Kd);

newValues = true;

server.setHeader("Cache-Control", "no-cache, no-store must-revalidate");
server.setHeader("Pragma", "no-cache");
server.setHeader("Expires", "-1");
server.setContentLength(0);
server.send(200, "text/plain", "");
server.sendContent("");
server.client().stop();
}

void SendFile (void) {
#if DEV_FILE
    Serial.println ("SEND FILE");
#endif
    File download = sd.open("/maqueta/" + filename);//, "r");
    if (download){
#if DEV_FILE
        Serial.println ("OBERT");
#endif
        server.setHeader("Content-Type", "text/text");
        server.setHeader("Content-Disposition", "attachment; filename=" + filename);
        server.setHeader("Connection", "close");
        server.streamFile(download, "application/octet-stream");
        server.sendContent("");
        server.client().stop();
        fileState = 0x00;
        download.close();
    }
    stateFile = 0x06;
}

void UpdateClient(void) {
/*SENDHTMLHEADER-----*/
    SendHTML_Header();
/*SENDHTMLCONTENT-----*/
    append_page_content();
    server.sendContent(webpage);
    webpage = "";
    server.sendContent("");

    server.client().stop(); // Stop is needed because no content length was sent
}

void SendHTML_Header(){
    server.setHeader("Cache-Control", "no-cache, no-store, must-revalidate");
    server.setHeader("Pragma", "no-cache");
    server.setHeader("Expires", "-1");
    server.setContentLength(CONTENT_LENGTH_UNKNOWN);
    server.send(200, "text/html", ""); // Empty content inhibits Content-length header so we have to close the socket ourselves.
    append_page_header();
    server.sendContent(webpage);
    webpage = "";
}

void UpdateFile (void) {
    static uint8_t fileExists = 0;

    switch (stateFile) {
        case 0x00: //Waiting order to start file
            if ((fileState) == 0x01) {
                if (fileExists = 1) stateFile = 0x11;
                else stateFile = 0x01;
            }
            break;

        case 0x11: //Delete File if already exists before open it again
            //sd.remove("/") + filename);
            stateFile = 0x01;
            break;
    }
}

```



```

    case 0x01: //Open File
        numFile++;
        filename = String(numFolder) + "/" + String(numFile) + ".txt";
        fileW = sd.open("/maqueta/" + filename, O_WRITE | O_CREAT);
        printHeader();
        millisFitxer = 0;
    #if DEV_FILE
        Serial.println ("OPEN FILE");
    #endif
        stateFile = 0x02;
        break;

    case 0x02: //Writing File
        if ((fileState) == 0x02) stateFile = 0x03;
        break;

    case 0x03: //Close File
    #if DEV_FILE
        Serial.println ("CLOSE FILE");
    #endif
        fileW.close();
        fileExists = 1;
        stateFile = 0x04;
        break;

    case 0x04: //Close File
    #if DEV_FILE
        Serial.println ("CLOSED FILE");
    #endif
        stateFile = 0x00;
        break;

    default:
        stateFile = 0x00;
        break;
    }
}
/*----- TIMER -----*/
void TimerInterrupt(void) {
    UpdateMagnets();
    if (stateFile == 0x02){
        printFile();
    }
    motorCalc();
}

void printFile(void){
    millisFitxer ++;

    if ((millisFitxer % 5) == 0){
        fileW.print(millisFitxer);
        fileW.print(",");
        fileW.print(Output);
        fileW.print(",");
        fileW.print(Input);
        fileW.print(",");
        fileW.print(sensorSetpoint);
        fileW.print(",");
        fileW.print(workSetpoint);
        fileW.print(",");
        fileW.print(Amps);
        fileW.print(",");
        fileW.print(mpu6050.getTemp());
        fileW.print(",");
        fileW.print(mpu6050.getAccX());
        fileW.print(",");
        fileW.print(mpu6050.getAccY());
        fileW.print(",");
        fileW.print(mpu6050.getAccZ());
        fileW.print(",");
        fileW.print(mpu6050.getGyroX());
        fileW.print(",");
        fileW.print(mpu6050.getGyroY());
        fileW.print(",");
        fileW.print(mpu6050.getGyroZ());
        fileW.print(",");
        fileW.print(mpu6050.getAccAngleX());
        fileW.print(",");
        fileW.print(mpu6050.getAccAngleY());
        fileW.print(",");
        fileW.print(mpu6050.getGyroAngleX());
        fileW.print(",");
        fileW.print(mpu6050.getGyroAngleY());
        fileW.print(",");
        fileW.print(mpu6050.getGyroAngleZ());
        fileW.print(",");
        fileW.print(mpu6050.getAngleX());
        fileW.print(",");
        fileW.print(mpu6050.getAngleY());
        fileW.print(",");
        fileW.print(mpu6050.getAngleZ());
        if (newValues){
            fileW.print(",");
            fileW.print(motorState);
            fileW.print(",");
            fileW.print(imantsState);
            fileW.print(",");
            fileW.print(tempsPols);
            fileW.print(",");
            fileW.print(rpm);
            fileW.print(",");
            fileW.print(Kp);
            fileW.print(",");
            fileW.print(Ki);

```

```

        fileW.print("");
        fileW.print(Kd);
        newValues = false;
    }
    fileW.println("");
}

num++;
if (num == 100) {
    fileW.flush();
    num = 0;
}
}
/*----- SENSORS (C) & IMANTS (L) -----*/
void UpdateMagnets(void) {
    static uint8_t writeByteOld2 = 0x00;
    static uint8_t readByteOld = 0x00;
    static uint8_t count[4] = {0,0,0,0};
    uint8_t tempWriteByte = 0x00, tempWriteByte2 = 0x00, writeByteOld = 0x00, newReadByte = 0x00;

#if DEV_MAG
    static uint16_t n1 = 0;
    n1++;
#endif

    digitalWrite(LATCH, LOW);
    if (imantsOn == true) {
        shiftOut(DATA_OUT, CLOCK, MSBFIRST, writeByte);
        writeByteOld2 = writeByte;
    } else{
        tempWriteByte2 = writeByte&0xC3;
        tempWriteByte2 |= 0x3C;
        shiftOut(DATA_OUT, CLOCK, MSBFIRST, tempWriteByte2);
    }
    digitalWrite(LATCH, HIGH);
    readByte = shiftIn(DATA_IN, CLOCK, MSBFIRST);
    // Per fer anar els 4 sensors

    if (tempsPols == 0x00){
        tempWriteByte = ~((readByte&0xF0)>>2);
        tempWriteByte &= 0x3C;
    }else{
        if(((readByte&0x80)==0x80)&&((readByteOld&0x80)==0x00)) {
            newReadByte |= 0x80;
            count[0] = tempsPols;
        } else{
            if(count[0]!=0){
                newReadByte |= 0x80;
                count[0]--;
            } else{
                newReadByte &= 0x7F;
            }
        }

        if(((readByte&0x40)==0x40)&&((readByteOld&0x40)==0x00)) {
            newReadByte |= 0x40;
            count[1] = tempsPols;
        } else{
            if(count[1]!=0){
                newReadByte |= 0x40;
                count[1]--;
            } else{
                newReadByte &= 0xBF;
            }
        }

        if(((readByte&0x20)==0x20)&&((readByteOld&0x20)==0x00)) {
            newReadByte |= 0x20;
            count[2] = tempsPols;
        } else{
            if(count[2]!=0){
                newReadByte |= 0x20;
                count[2]--;
            } else{
                newReadByte &= 0xDF;
            }
        }

        if(((readByte&0x10)==0x10)&&((readByteOld&0x10)==0x00)) {
            newReadByte |= 0x10;
            count[3] = tempsPols;
        } else{
            if(count[3]!=0){
                newReadByte |= 0x10;
                count[3]--;
            } else{
                newReadByte &= 0xEF;
            }
        }

        tempWriteByte = ~((newReadByte&0xF0)>>2);
        tempWriteByte &= 0x3C;
    }
    readByteOld = readByte;

    writeByte &= 0xC3;
    writeByte |= tempWriteByte;

#if DEV_MAG
    if ((n1%1000) == 0){
        Serial.print ("Read: ");
        Serial.print (readByte, HEX);
        Serial.print (" - Temp: ");
        Serial.print (tempWriteByte, HEX);
    }
}

```

```

        Serial.print (" - Write: ");
        Serial.println (writeByte, HEX);
    }
#endif
}

/*----- SENSOR CURRENT -----*/
void ReadCurrent(void) {
    uint8_t index = 0;
    double MeanVoltage = 0;
    double VoltageRead = 0, Voltage = 0;

    for (index = 0; index < N_READS; index++){
        VoltageRead = (esp_adc_cal_raw_to_voltage(adc1_get_raw(ADC1_CHANNEL_0), &adc_chars));
        Voltage = (VoltageRead-300.0)*2.5575;
        MeanVoltage += Voltage;
    }
    MeanVoltage /= N_READS;

    Amps = ((MeanVoltage - (ACS_OFFSET)) / MV_PER_AMP);

#if DEV_CUR
    int RawVoltage = adc1_get_raw(ADC1_CHANNEL_0);
    Serial.print("RawVoltage = "); // shows pre-scaled value
    Serial.print(RawVoltage); // the '3' after voltage allows you to display 3 digits after decimal point
    Serial.print("\t VoltageRead = "); // shows pre-scaled value
    Serial.print(VoltageRead, 3); // the '3' after voltage allows you to display 3 digits after decimal point
    Serial.print("\t Voltage = "); // shows pre-scaled value
    Serial.print(MeanVoltage, 3); // the '3' after voltage allows you to display 3 digits after decimal point
    Serial.print("\t Amps = "); // shows the voltage measured
    Serial.println(Amps, 3); // the '3' after voltage allows you to display 3 digits after decimal point
    delay(1000);
#endif
}

#if DEV_ACC
void mpu6050Test(void) {
    static long timer = 0;
    if (millis() - timer > 1000) {
        Serial.println("=====");
        Serial.print("temp : "); Serial.println(mpu6050.getTemp());
        Serial.print("accX : "); Serial.print(mpu6050.getAccX());
        Serial.print("\taccY : "); Serial.print(mpu6050.getAccY());
        Serial.print("\taccZ : "); Serial.println(mpu6050.getAccZ());

        Serial.print("gyroX : "); Serial.print(mpu6050.getGyroX());
        Serial.print("\tgyroY : "); Serial.print(mpu6050.getGyroY());
        Serial.print("\tgyroZ : "); Serial.println(mpu6050.getGyroZ());

        Serial.print("accAngleX : "); Serial.print(mpu6050.getAccAngleX());
        Serial.print("\taccAngleY : "); Serial.println(mpu6050.getAccAngleY());

        Serial.print("gyroAngleX : "); Serial.print(mpu6050.getGyroAngleX());
        Serial.print("\tgyroAngleY : "); Serial.print(mpu6050.getGyroAngleY());
        Serial.print("\tgyroAngleZ : "); Serial.println(mpu6050.getGyroAngleZ());

        Serial.print("angleX : "); Serial.print(mpu6050.getAngleX());
        Serial.print("\tangleY : "); Serial.print(mpu6050.getAngleY());
        Serial.print("\tangleZ : "); Serial.println(mpu6050.getAngleZ());
        Serial.println("=====\\n");
        timer = millis();
    }
}
#endif

void initSD(void){
    char FolderName[16];
    uint16_t maxNumFolder = 0;
    if (!sd.begin(SD_CHIP_SELECT, SD_SCK_MHZ(10))) {
#if DEV_SD
        Serial.println("initialization failed!");
#endif
        return;
    }
    if DEV_SD
        Serial.println("initialization done.");
    #endif

    // Creem o Obrim carpeta /maqueta
    if (mainRoot.open("/Maqueta/")){
#if DEV_FILE
        Serial.println ("carpeta oberta");
#endif
    } else {
        if (sd.mkdir("Maqueta")){
#if DEV_FILE
            Serial.println ("Carpeta creada");
#endif
            if(mainRoot.open("/Maqueta/")){
                Serial.println("Carpeta oberta");
            }
        }else return;
    }

    //Busquem última carpeta dins seu i en creem una nova conseqüentiva
    while(newRoot.openNext(&mainRoot, O_RDONLY)){
        newRoot.getName(FolderName, 16);
        if DEV_SD
            Serial.print("Folder : ");
            newRoot.printName(&Serial);
            Serial.println();
        #endif
    }
}

```

```

        if(newRoot.isDir()){
            numFolder = String(FolderName).toInt();
            if (maxNumFolder<numFolder) maxNumFolder = numFolder;
        }
        newRoot.close();
    }
    numFolder = maxNumFolder+1;
    String Folder = "/Maqueta/"+String(numFolder);

    Folder.toCharArray(FolderName,16);
    if(sd.mkdir(FolderName)){
#ifdef DEV_FILE
        Serial.print("New folder created: ");
        Serial.println(FolderName);
#endif
    }
}

void printHeader(void){
    fileW.print("millis");
    fileW.print(";");
    fileW.print("Output");
    fileW.print(";");
    fileW.print("Input");
    fileW.print(";");
    fileW.print("Setpoint");
    fileW.print(";");
    fileW.print("WorkSetpoint");
    fileW.print(";");
    fileW.print("Current");
    fileW.print(";");
    fileW.print("Temperatura");
    fileW.print(";");
    fileW.print("AccX");
    fileW.print(";");
    fileW.print("AccY");
    fileW.print(";");
    fileW.print("AccZ");
    fileW.print(";");
    fileW.print("GyroX");
    fileW.print(";");
    fileW.print("GyroY");
    fileW.print(";");
    fileW.print("GyroZ");
    fileW.print(";");
    fileW.print("AccAngleX");
    fileW.print(";");
    fileW.print("AccAngleY");
    fileW.print(";");
    fileW.print("GyroAngleX");
    fileW.print(";");
    fileW.print("GyroAngleY");
    fileW.print(";");
    fileW.print("GyroAngleZ");
    fileW.print(";");
    fileW.print("AngleX");
    fileW.print(";");
    fileW.print("AngleY");
    fileW.print(";");
    fileW.print("AngleZ");

    fileW.print(";");
    fileW.print("Motor");
    fileW.print(";");
    fileW.print("Imants");
    fileW.print(";");
    fileW.print("Temps Pols");
    fileW.print(";");
    fileW.print("Setpoint");
    fileW.print(";");
    fileW.print("Kp");
    fileW.print(";");
    fileW.print("Ki");
    fileW.print(";");
    fileW.print("Kd");
}

```